UNIVERSIDAD DEL ACONCAGUA

FACULTAD DE CIENCIAS SOCIALES Y ADMINISTRATIVAS

LICENCIATURA EN INFORMÁTICA

PRÁCTICA PROFESIONAL

IMPLEMENTACIÓN DE NIVEL ADICIONAL DE SEGURIDAD INFORMÁTICA PARA AUTENTICAR USUARIOS.

Tesina para optar al grado de Licenciado en Informática y Desarrollo de Software

Autor: TÉCNICO EN INFORMÁTICA Y DESARROLLO DE SOFTWARE

MATÍAS SEBASTIÁN SOSA

Legajo: 17.769

Tutor: MAG. LIC. ALEJANDRO VAZQUEZ

Mendoza, junio de 2015

CALIFICACIÓN

ÍNDICE

RESUMEN TÉCNICO

La presente Tesina plantea el desarrollo de un nivel de seguridad extra a un conocido mecanismo de Seguridad Informática denominado *login*. Se seleccionan una serie de herramientas, las más adecuadas y actualizadas posibles para la implementación de un proyecto de este tipo, tales como el tipo de lenguaje, frameworks o marcos de trabajo y base de datos. La Tesina contará con un ejemplo práctico que se cristalizara en un desarrollo en el lenguaje *Java*, en su ámbito web con el empleo de *frameworks*, como *MyBatis* para desarrollar la persistencia, *Java Server Face* para el manejo y gestión de toda la parte web y un algoritmo propio desarrollado especialmente para este trabajo enfocado en el HASH que generará el sistema como clave. También se utilizará *MySql* como sistema de gestión de base datos y *NetBeans* 8.0.1 para gestionar el proyecto.

El objetivo de la tesina será mostrar en un ejemplo práctico el funcionamiento del mecanismo de login con la llave para acceder a él. Para lograrlo una persona ingresará usuario y contraseña. Luego de ser convalidados los datos, el sistema generará una llave (un HASH en SHA-1, pudiendo utilizarse en el futuro otros métodos alternativos que lo perfeccionen o esten más actualizados a la hora de su implementación) a partir de una serie de parámetros. Esta clave le será enviada al ingresante vía mail, con un asunto y un mensaje predeterminado y estandarizado en la aplicación. Posteriormente se le solicitará para ser validada.

El documento además cuenta con información adicional sobre metodologías y descripciones sobre leyes, criptografía y conceptos de lo que es la gran e interminable tarea de la Seguridad Informática.

1. INTRODUCCIÓN

Las organizaciones cuentan con sistemas que generalmente se caracterizan por capturar grandes cantidades de datos necesarios para registrar las operaciones que sostienen a una empresa.

Gran parte de esos datos son de carácter confidencial e inclusive sensible, al funcionamiento de la institución.

El constante crecimiento de los volúmenes de los datos lleva cada vez más a pensar en que protegerlos puede llegar a ser un arte y no una técnica, que derivan también del fuerte crecimiento de las tecnologías y de las nuevas necesidades que constantemente surgen día a día en el mundo.

Estamos inmersos en un entorno en el que la competitividad, la globalización, la consolidación de industrias, y un largo etcétera, hacen que la información juegue cada vez más un papel crítico.

Los datos referentes a mercados, competidores, clientes, inclusive a los indicadores de rendimiento de las propias organizaciones, se han convertido en un recurso clave. Si no son protegidos de la manera adecuada, puede convertirse en un hito para la empresa, siendo perjudicial o beneficioso para la misma y pudiendo marcar un rumbo en las pérdidas o ganancias, imagen institucional, entre muchas cuestiones por nombrar.

La no protección de los datos puede llegar a convertirse en un karma a la hora de interactuar con un usuario, si no se poseen los recaudos y resguardos necesarios para trabajar con ese tipo de información. Se puede llegar a comprometer y perjudicar a muchas personas que solo querían utilizar un servicio de una aplicación determinada y se vieron inmersos en un mundo de problemas legales y muchas veces económicos. Siendo cada persona propietaria del dato al notar anormalidades, difusión, modificaciones o pérdida de sus datos, esto, va a generar un malestar general y una muy mala fama para la institución en la que se confió esa información. E incluso generar juicios con pérdidas cuantiosas para la institución, sea ésta pública o privada.

Por tal motivo, se debe hacer especial hincapié en salvaguardar la información sensible del usuario de una manera precavida y hasta heroica ya que, por ejemplo, para un usuario, que se difunda su número de tarjeta o reciba gastos que él no aprobó, puede concluir en problemas legales, económicos y de reputación muy graves para la institución e inclusive llevar a que la credibilidad que la misma aporta a la comunidad se caiga a pedazos.

Alguno de los tantos métodos de seguridad e inclusive el más utilizado que pueden existir dentro de las diferentes aplicaciones y sobre todo aplicaciones o sistemas que corren en la web, es el “login” (o registro de ingreso). Este procedimiento consta de un proceso de validación de un usuario y una contraseña, ambos elegidos por el usuario (salvo casos excepcionales) en donde luego de ser ingresados y ratificados contra la base de datos se procede a acceder al sistema y realizar el usufructo del mismo.

Las personas o empresas que poseen una clave y un usuario del sistema, ya sea de escritorio o web, son aquellas que desean hacer uso de las funcionalidades que el sistema ofrece. Lo que pretenden con ese mecanismo es agregar un nivel más de seguridad para logar proteger la privacidad, integridad y confidencialidad de sus datos y a su vez garantizar la legítima disponibilidad de los mismos a su auténtico dueño.

Existen aplicaciones, o también llamados sistemas web, como redes sociales, que permiten una breve configuración de la privacidad de los datos y su visibilidad a otros usuarios, pero interiormente, los datos deben estar correctamente protegidos y con un énfasis criterioso al momento de su resguardo y respaldo.

Generalmente, las empresas utilizan diferentes formas de cuidar los datos, ya sea con software de terceros o con desarrollos propios. Asimismo, las personas se resguardan la información a mostrar y crean identidades falsas, mails paralelos para distintas cuentas dentro de los sistemas (web sobre todo) y muchas de estas diferentes y criteriosas soluciones son recomendadas por especialistas y sitios web.

Del lado de la organización se utilizan técnicas de seguridad, como métodos hash (encriptación unidireccional), encriptación simple, zonas desmilitarizadas y muchas otras técnicas o artilugios informáticos para lograr proteger así, la información respaldada en una base de datos. Amén de los sistemas de backup y recupero de ellos, que forman parte insoslayable de cualquier sistema informático con pretensiones de seriedad.

1.1. Problema de investigación

Como se describió anteriormente en los sistemas, sobre todo en ambientes web, la seguridad se convierte en un problema ya que se trabaja con datos ultrasensibles, como datos de tarjetas de crédito o débito.

Aunque los sistemas de seguridad tradicionales solucionan muchos de estos problemas, nunca es suficiente para proteger los datos, ya que cualquier *backing door* (puerta trasera o agujero de seguridad) puede convertir en un caos la organización.

Otro de los problemas de seguridad que se plantea es el de permisos de usuarios. El punto crítico de ellos son las cuentas de usuarios invitados y/o usuarios por defecto. Las buenas prácticas universalmente aceptadas sugieren que se eliminen debido a que dejan brechas de seguridad en las redes e inclusive en el mismo sistema.

Un usuario por defecto, por ejemplo *root, admin, manager*, etc., debería ser deshabilitado, inhabilitado o inclusive eliminado del sistema (de ser posible), ya que generalmente son súper usuarios en cuanto a privilegios y algún usuario de ese tipo puede ser un enemigo a la hora de cuidar datos de usuarios y/o de la organización.

Sumado a los problemas generales de seguridad, también se incluyen uno de los más peligrosos que se concibe en las organizaciones que es el de la Ingeniería Social, el método por el cual se puede extraer información de diversos modos a los usuarios. El hecho de prestar claves, usuarios, hacer favores, entre otras cuestiones irremediables son un problema para la empresa. Este tipo de amenaza no se puede eliminar, pero se puede mitigar ligando todos sus usuarios a un usuario único y unificado para el manejo de todos los sistemas –incluyendo mail-. Por lo tanto, si una persona presta su usuario y contraseña a otra, se ven comprometidos todos sus datos, participación y responsabilidad en la empresa, Como consecuencia, el propietario será responsable de todo lo que lo que ocurra con su cuenta “oficial”.

Al mirar desde otro punto de vista, también se encuentra la negligencia de muchos empleados al no bloquear sus equipos, no actualizar los sistemas operativos, ni sus aplicaciones. En general, si este proceso no es automático y aunque este fuera automático, tampoco se realizan porque los empleados en sus puestos de trabajo generalmente tienden a cancelarlo porque creen les resulta innecesario o también cargoso y molesto para el usuario. Siempre en las organizaciones, los encargados de la seguridad deben hacer un esfuerzo extra para lograr capacitar al personal y hacerle entender la criticidad de estos hechos o lograr mecanismos eficientes, probados, con cierto nivel estable de seguridad y que funcionen en modo

“silencioso” (oculto al usuario final) para garantizar las actualizaciones del sistema operativo y programas cruciales que utilicen los empleados en sus terminales.

Los *malware*, en sus tantas formas conocidas como troyanos, virus, *spyware, ransomware, adaware*, entre otros tantos tipos de variaciones que existen, una vez que se manifiestan en las computadoras de los usuarios suelen causar pánico, y al sentirse amenazados entran en apuros de querer salvar su información, y no saber si son espiados, si perdió datos, se los robaron o tiene algún archivo infectado.

Con el fin de concientizar sobre todos estos temas y otros no mencionados, se debe hacer gran hincapié en la seguridad, ya que hoy en día gran porcentaje de la información personal está en la red. Asimismo, en los últimos tiempos, con el espionaje existente por parte de las organizaciones y gobiernos locales y/o extranjeros inclusive, se tiene que seleccionar conscientemente la información personal que realmente queremos compartir.

1.2. Objetivos

Lo que se pretende lograr con este trabajo de Tesina es implementar niveles automatizados adicionales de seguridad para autenticar a usuarios que permitirán principalmente:

- **ACCESO ÚNICO AL SISTEMA:** Puede ingresar al sistema toda aquella persona que valide con usuario, contraseña y que ingrese la clave encriptada que se le envió al correo personal convenientemente registrado en la base de datos.

- **AGREGAR UNA CAPA EXTRA DE AUTENTICACIÓN A LA SEGURIDAD DEL SISTEMA:** Agregar al método más utilizado de seguridad, el *login*, un nivel más de seguridad que permita elevar la confiabilidad en él y su uso.

- **ENCRIPTAR LA CONTRASEÑA ENVIADA AL CORREO ELECTRÓNICO.** La clave que se enviará al mail estará cifrada con *SHA-1*.

Para lograr estos objetivos el trabajo contemplará:

- **EJEMPLO PRÁCTICO:** Se realizará con un ejemplo breve y conciso, de una problemática común, un sistema de envergadura chica donde se acceda por *login* empleando la metodología planteada.

- **EMPLEO DE *FRAMEWORKS*:** El ejemplo citado anteriormente se implementará utilizando algunos *frameworks*[1] para ampliar el desarrollo y facilitar así la programación de la aplicación.

1.3. Justificación

La implementación de un nivel más de seguridad brindará, tanto al usuario como a la organización, la tranquilidad de que sus datos y su sesión no puedan ser utilizados por otra persona.

Para lograr el objetivo de elevar el nivel de seguridad planteado, se recurre a la utilización de un sistema combinado de claves que funcionan como llaves de seguridad del aplicativo.

Con el fin de satisfacer las necesidades cambiantes de las personas a medida que avanza la tecnología, resulta imprescindible ir reforzando, ampliando y sobretodo renovando los métodos que incumben a la seguridad.

Para cumplir con el punto planteado anteriormente, se propone el desarrollo de una extensión de seguridad a los sistemas de login convencionales que, consta de enviar vía correo electrónico una clave ÚNICA generada al momento de iniciar la utilización del sistema con la que el usuario, además de los datos de login deberá utilizar esta clave para poseer el acceso a las funcionalidades. Esta clave posee un tiempo de caducidad que al momento de vencerse o de cerrar sesión, la llave no sirve para una sesión posterior.

A partir de la seguridad de los datos y sobre todo de la seguridad sobre las transacciones que pueda llegar a utilizar la organización, la confianza del usuario final para concretizar y utilizar sus datos verídicos irán creciendo. A través del tiempo esto se fue mejorando y las personas que poseían temor a ingresar por ejemplo su número de tarjeta de crédito a un sistema web o página web, fue reduciéndose a tal punto que en la actualidad se pagan todo tipo de impuestos, transferencias bancarias, entre otras por mencionar solo algunas de las operaciones por internet.

---

[1] En el desarrollo de *software*, un ***framework*** o ***infraestructura digital***, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, que puede servir de base para la organización y desarrollo de *software*. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. Fuente: http://es.wikipedia.org/wiki/Framework

1.4. Organización del documento

El presente documento está organizado en diferentes capítulos, los que se describen de la siguiente manera:

Capítulo 1: Introducción

Se realiza una breve reseña de lo que se busca desarrollar a lo largo de este escrito y se comentan problemáticas generales de las organizaciones.

Capítulo 2: Marco Conceptual

Se describen la tecnología asociado a lo que es la seguridad informática en general, mecanismos de seguridad y encriptación.

Capítulo 3: Desarrollo de la Tesina

En este capítulo se describen los problemas que resolverá la propuesta de desarrollo. El objetivo de este capítulo es asegurar que se reúnan, se comprendan, se documenten y se autoricen los requerimientos antes de dar comienzo formal a la implementación. Asimismo, describe la arquitectura del sistema.

Diseño y documentación del sistema que se utilizará de ejemplo en la tesina.

Justificación de herramientas utilizadas.

Justificación de tecnologías a utilizar.

Capítulo 4: Conclusiones

Se describen las conclusiones obtenidas de la Tesina.

Capítulo 5: Bibliografía

Se describe la bibliografía utilizada y todos los sitios de internet.

2. MARCO CONCEPTUAL

Gran parte del marco conceptual es resumido del libro: Seguridad Informática de Purificación Aguilera López, editorial Editex S.A.

2.1. Sistemas de información y sistemas informáticos

Un sistema de información (SI) es un conjunto de elementos organizados, relacionados y coordinados entre sí, encargados de facilitar el funcionamiento global de una empresa o de cualquier otra actividad humana para conseguir sus objetivos.

Estos elementos son:

- Recursos. Pueden ser físicos, como ordenadores, componentes, periféricos y conexiones, recursos no informáticos; y lógicos, como sistemas operativos y aplicaciones informáticas.

- Equipo humano. Compuesto por las personas que trabajan para la organización.

- Información. Conjunto de datos organizados que tienen un significado. La información puede estar contenida en cualquier tipo de soporte.

- Actividades que se realizan en la organización, relacionadas o no con la informática.

2.1.1.Sistemas informáticos

Un sistema informático está constituido por un conjunto de elementos físicos (*hardware*, dispositivos, periféricos y conexiones), lógicos (sistemas operativos, aplicaciones, protocolos...) y con frecuencia se incluyen también los elementos humanos (personal experto que maneja el *software* y el *hardware*).

2.1.2.Actividad en un sistema informático

Un sistema informático puede ser un subconjunto del sistema de información, pero en principio un sistema de información no tiene por qué contener elementos informáticos, aunque en la actualidad es difícil imaginar cualquier actividad humana en la que no se utilice la informática. A lo largo de este libro estudiaremos la seguridad en los sistemas de información, en general, y en los sistemas informáticos, en particular, como parte de aquellos.

2.1.3.Seguridad

2.1.3.1.Aproximación al concepto de seguridad en sistemas de información

Una de las acepciones de la RAE para el término “seguro”, que es la que aquí nos interesa, es la de estar libre y exento de todo peligro, daño o riesgo. Este es el concepto en el que se basa el

contenido de este libro y tiene el mismo sentido aplicado a sistemas de información y sistemas informáticos.

La seguridad informática es la disciplina que se ocupa de diseñar las normas, procedimientos, métodos y técnicas destinados a conseguir un sistema de información seguro y confiable.

Un sistema de información, no obstante las medidas de seguridad que se le apliquen, no deja de tener siempre un margen de riesgo.

Para afrontar el establecimiento de un sistema de seguridad es necesario conocer:

- Cuáles son los elementos que componen el sistema. Esta información se obtiene mediante entrevistas con los responsables o directivos de la organización para la que se hace el estudio de riesgos y mediante apreciación directa.

- Cuáles son los peligros que afectan al sistema, accidentales o provocados. Se deducen tanto de los datos aportados por la organización como por el estudio directo del sistema mediante la realización de pruebas y muestreos sobre él.

- Cuáles son las medidas que deberían adoptarse para conocer, prevenir, impedir, reducir o controlar los riesgos potenciales. Se trata de decidir cuáles serán los servicios y mecanismos de seguridad que reducirían los riesgos al máximo posible.

Tras el estudio de riesgos y la implantación de medidas, debe hacerse un seguimiento periódico, revisando y actualizando las medidas adoptadas.

Todos los elementos que participan en un sistema de información pueden verse afectados por fallos de seguridad, aunque se suele considerar la información como el factor más vulnerable. El *hardware* y otros elementos físicos se pueden volver a comprar o restaurar, el *software* puede ser reinstalado, pero la información dañada no siempre es recuperable, lo que puede ocasionar daños de diversa índole sobre la economía y la imagen de la organización y, a veces, también causar perjuicios a personas. Otro aspecto a tener en cuenta es que la mayoría de los fallos de seguridad se deben al factor humano.

2.1.3.2.Tipos de seguridad

2.1.3.2.1.Activa

Comprende el conjunto de defensas o medidas cuyo objetivo es evitar o reducir los riesgos que amenazan al sistema.

Ejemplos: impedir el acceso a la información a usuarios no autorizados mediante introducción de nombres de usuario y contraseñas; evitar la entrada de virus instalando un antivirus; impedir, mediante encriptación, la lectura no autorizada de Mensajes.

2.1.3.2.2.Pasiva

Está formada por las medidas que se implantan para, una vez producido el incidente de seguridad, minimizar su repercusión y facilitar la recuperación del sistema; por ejemplo, teniendo siempre al día copias de seguridad de los datos.

2.1.3.3.Propiedades de un sistema de información seguro

Los daños producidos por falta de seguridad pueden causar pérdidas económicas o de credibilidad y prestigio a una organización.

Su origen puede ser:

- Fortuito. Errores cometidos accidentalmente por los usuarios, accidentes, cortes de fluido eléctrico, averías del sistema, catástrofes naturales...
- Fraudulento. Daños causados por *software* malicioso, intrusos o por la mala voluntad de algún miembro del personal con acceso al sistema, robo o accidentes provocados.

Se considera seguro un sistema que cumple con las propiedades de integridad, confidencialidad y disponibilidad de la información. Cada una de estas propiedades conlleva la implantación de determinados servicios y mecanismos de seguridad que se estudiarán más adelante.

2.1.3.3.1.Integridad

Este principio garantiza la autenticidad y precisión de la información sin importar el momento en que esta se solicita, o dicho de otra manera, una garantía de que los datos no han sido alterados ni destruidos de modo no autorizado.

Para evitar este tipo de riesgos, se debe dotar al sistema de mecanismos que prevengan y detecten cuándo se produce un fallo de integridad y que puedan tratar y resolver los errores que se han descubierto.

2.1.3.3.2.Confidencialidad

La OCDE (Organización para la Cooperación y el Desarrollo Económico), en sus Directrices para la Seguridad de los Sistemas de Información define la confidencialidad como “el hecho de que los datos o informaciones estén únicamente al alcance del conocimiento de las personas, entidades o mecanismos autorizados, en los momentos autorizados y de una manera autorizada”.

Para prevenir errores de confidencialidad, debe diseñarse un control de accesos al sistema: quién puede acceder, a qué parte del sistema, en qué momento y para realizar qué tipo de operaciones.

2.1.3.3.3.Disponibilidad

La información ha de estar disponible para los usuarios autorizados cuando la necesiten.

Las normativas MAGERIT definen la disponibilidad como "grado en el que un dato está en el lugar, momento y forma en que es requerido por el usuario autorizado". Situación que se produce cuando se puede acceder a un sistema de información en un periodo de tiempo considerado aceptable. La disponibilidad está asociada a la fiabilidad técnica de los componentes del sistema de información.

Se deben aplicar medidas que protejan la información, así como crear copias de seguridad y mecanismos para restaurar los datos que accidental o intencionadamente se hubiesen dañado o destruido.

2.2. Análisis de Riesgos

A la hora de dotar de seguridad a un sistema de información, hay que tener en cuenta todos los elementos que lo componen, analizar el nivel de vulnerabilidad de cada uno de ellos ante determinadas amenazas y valorar el impacto de un ataque que causaría sobre todo el sistema.

La persona o el equipo encargado de la seguridad deberán analizar con esmero cada uno de los elementos. A veces el descuido de un elemento considerado débil ha producido importantes fallos de seguridad. Al estar interrelacionados todos los elementos, este descuido puede producir errores en cadena con efectos insospechados sobre la organización.

2.2.1.Elementos de estudio

Para comenzar a analizar un sistema de información al que se pretende dotar de unas medidas de seguridad, hay que tener en cuenta los siguientes elementos: activos, amenazas, riesgos, vulnerabilidades, ataques e impactos.

2.2.1.1.Activos

Son los recursos que pertenecen al propio sistema de información o que están relacionados con este. La presencia de activos facilita el funcionamiento de la empresa u organización y la consecución de sus objetivos. Al hacer un estudio de los activos existentes, hay que tener en cuenta la relación que guardan entre ellos y la influencia que se ejercen: cómo afectaría en uno de ellos un daño ocurrido a otro.

Podemos clasificarlos en los siguientes tipos:

2.2.1.2.Datos

Los datos constituyen el núcleo de toda la organización, hasta el punto que se tiende a considerar que el resto de los activos están al servicio de la protección de los datos. Normalmente están organizados en bases de datos y almacenados en soportes de diferente tipo. El funcionamiento de una empresa y organización depende de sus datos, que pueden ser de todo tipo: económicos, fiscales, de recursos humanos, clientes o proveedores.

Cada tipo de dato merece un estudio independiente de riesgo por la repercusión que su deterioro o pérdida pueda causar, como por ejemplo los relativos a la intimidad y honor de las personas y otros de índole confidencial.

2.2.1.3.Software

El software constituido por los sistemas operativos y el conjunto de aplicaciones instaladas en los equipos de un sistema de información que reciben y gestionan o transforman los datos para darles el fin que se tenga establecido.

2.2.1.4.Hardware

Se trata de los equipos (servidores y terminales) que contienen las aplicaciones y permiten su funcionamiento, a la vez que almacenan los datos del sistema de información. Incluimos en este grupo los periféricos y elementos accesorios que sirven para asegurar el correcto funcionamiento de los equipos o servir de vía de transmisión de los datos (módem, router, instalación eléctrica o sistemas de alimentación ininterrumpida, destructores de soportes informáticos).

2.2.1.5.Redes

Desde las redes locales de la propia organización hasta las metropolitanas o internet. Representan la vía de comunicación y transmisión de datos a distancia

2.2.1.6.Soportes

Los lugares en donde la información queda registrada y almacenada durante largos períodos o de forma permanente (DVD, CD, tarjetas de memoria, discos duros externos dedicados al almacenamiento, microfilms e incluso papel)

2.2.1.7.Instalaciones

Son los lugares que albergan los sistemas de información y de comunicaciones. Normalmente se trata de oficinas, despachos, locales o edificios, pero también pueden ser vehículos y otros medios de desplazamiento.

2.2.1.8.Personal

Es el conjunto de personas que interactúan con el sistema de información: administradores, programadores, usuarios internos y externos y resto de personal de la empresa. Los estudios calculan que se producen más fallos de seguridad por intervención del factor humano que por fallos en la tecnología.

2.2.1.9.Servicios

Servicios que se ofrecen a clientes o usuario: productos, servicios, sitios web, foros, correo electrónico y otros servicios de comunicaciones, información, seguridad, etc.

2.2.2.Amenazas

En sistemas de información se entiende por amenaza a la presencia de uno o más factores de diversa índole (personas, máquinas o sucesos) que – de tener la oportunidad – (atacarían al sistema produciéndole daños aprovechándose de su nivel de vulnerabilidad). Hay diferentes tipos de amenazas de las que hay que proteger al sistema. Por un lado existen amenazas físicas como cortes eléctricos, fallos del *hardware* o riesgos ambientales hasta los errores intencionados o no de los usuarios, la entrada de *software* malicioso (virus, troyano, gusanos} o el robo, destrucción o modificación de la información.

En función del tipo de alteración, daño o interrupción que podrían producir sobre la información, las amenazas se clasifican en cuatro grupos:

2.2.2.1.De interrupción

El objetivo de la amenaza es deshabilitar el acceso a la información, por ejemplo destruyendo componentes físicos como el disco duro, bloqueando el acceso a los datos, o cortando o saturando los canales de comunicación.

2.2.2.2.De interceptación

Personas, programas o equipos no autorizados podrían acceder a un determinado recurso del sistema y captar información confidencial de la organización, como pueden ser datos, programas o identidad de personas.

2.2.2.3.De modificación

Personas, programas, o equipos no autorizados no solamente accederían a los programas y datos de un sistema de información, sino que además los modificarían. Por ejemplo, modificar la respuesta enviada a un usuario conectado o alterar el comportamiento de una aplicación instalada.

2.2.2.4.De fabricación

Las amenazas de fabricación agregarían información falsa en el conjunto de información del sistema.

Según su origen las amenazas se clasifican en:

2.2.2.4.1.Accidentales

Las amenazas accidentales pueden ser accidentes meteorológicos, incendios, inundaciones, fallos en los equipos, en las redes, en los sistemas operativos o en el *software*, errores humanos.

2.2.2.4.2.Intencionadas

Son debidas siempre a la acción humana, como la introducción de *software* malicioso – *malware* – (aunque este penetre en el sistema por algún procedimiento automático, su origen es siempre humano), intrusión informática (con frecuencia se produce previa la introducción de *malware* en los equipos), robos o hurtos. Las amenazas intencionadas pueden tener su origen en el exterior de la organización o incluso en su personal.

2.2.3.Riesgos

Se denomina riesgo a la posibilidad de que se materialice o no una amenaza aprovechando una vulnerabilidad. Una amenaza no constituye riesgo cuando no hay vulnerabilidad, ni una vulnerabilidad cuando no existe amenaza.

Ante un determinado riesgo, una organización puede optar por tres alternativas distintas:

- Asumirlo sin hacer nada. Esto solamente resulta lógico cuando el perjuicio esperado no tiene valor alguno o cuando el coste de aplicación de medidas superaría al de la reparación del daño.

- Aplicar medidas para disminuirlo o anularlo.

- Transferirlo (por ejemplo, contratando un seguro).

2.2.4.Vulnerabilidades

Probabilidades que existen de que una amenaza se materializa contra un activo. No todos los activos son vulnerables a las mismas amenazas. Por ejemplo, los datos son vulnerables a la acción de los *hackers*, mientras que una instalación eléctrica es vulnerable a un cortocircuito. Al hacer el análisis de riesgos hay que tener en cuenta la vulnerabilidad de cada activo.

2.2.5.Ataques

Se dice que se ha producido un ataque accidental o deliberado contra el sistema cuando se ha materializado una amenaza.

En función del impacto causado a los activos atacados, los ataques se clasifican de la siguiente manera:

2.2.5.1.Activos

Son denominados ataques activos si modifican, dañan, suprimen o agregan información o bien bloquean o saturan los canales de comunicación.

2.2.5.2.Pasivos

Son denominados ataques pasivos solamente a aquellos acceden sin autorización a los datos contenidos en el sistema. Son los más difíciles de detectar.

Un ataque puede ser directo o indirecto, si se produce desde el atacante al elemento “víctima” directamente o a través de recursos o personas intermediarias.

2.2.5.3.Impactos

Son la consecuencia de la materialización de una o más amenazas sobre uno o varios activos aprovechando la vulnerabilidad del sistema, dicho de otra manera, el daño causado.

Los impactos pueden ser cuantitativos, si los perjuicios pueden cuantificarse económicamente o cualitativos, si suponen daños no cuantificables, como los causados contra los derechos fundamentales de las personas.

2.2.6.Control de Riesgos

Una vez que se ha realizado el análisis de riesgos, se tiene que determinar cuáles serán los servicios necesarios para conseguir un sistema de información seguro. Para poder dar esos servicios será necesario dotar al sistema de los mecanismos correspondientes.

2.3. Servicios de Seguridad

2.3.1.Integridad

Asegurar que los datos del sistema no han sido alterados ni cancelados por personas o entidades no autorizadas y que el contenido de los mensajes recibidos es el correcto.

2.3.1.1.Confidencialidad

Proporcionar protección contra la revelación deliberada o accidental de los datos en una comunicación.

2.3.1.2.Disponibilidad

Permitirá que la información esté disponible cuando lo requieran las entidades autorizadas.

2.3.1.3.Autenticación (o identificación)

El sistema debe ser capaz de verificar que un usuario identificado que accede al sistema o que genera una determinada información es quien dice ser. Solo cuando un usuario o entidad ha sido autenticado, podrá tener autorización de acceso. Se puede exigir autenticación en la entidad de origen de la información, en la de destino o en ambas.

2.3.1.4.No repudio (o irrenunciabilidad)

Proporcionará al sistema una serie de evidencias irrefutables de la autoría de un hecho.

El no repudio consiste en no poder negar haber emitido una información que sí se emitió y en no poder negar su recepción cuando sí ha sido recibida

De esto se deduce que el no repudio puede darse de la siguiente manera:

2.3.1.4.1.En origen

El emisor no puede negar el envío porque el receptor tiene pruebas certificadas del envío y de la identidad del emisor. Las pruebas son emitidas por el propio emisor.

2.3.1.4.2.En destino

En este caso es el destinatario quien no puede negar haber recibido el envío ya que el emisor tiene pruebas infalsificables del envío y de la identidad del destinatario. Es el receptor quien crea las pruebas.

2.3.1.4.3.Control de acceso

Podrán acceder a los recursos del sistema solamente el personal y usuarios con autorización

2.4. Mecanismos de Seguridad

Según la función que desempeñen los mecanismos de seguridad pueden clasificarse de la siguiente manera:

2.4.1.Preventivos

Actúan antes de que se produzca un ataque, su misión es evitarlo.

2.4.2.Detectores

Actúan cuando el ataque se ha producido y antes de que cause daños en el sistema.

2.4.3.Correctores

Actúan después de que haya habido un ataque y se hayan producido daños. Su misión es la de corregir las consecuencias del daño.

Cada mecanismo ofrece al sistema uno o más servicios de los especificados anteriormente.

Existen muchos y variados mecanismos de seguridad. En esta sección se mencionan los más habituales, que se detallarán en otras didácticas.

La elección de mecanismos de seguridad depende de cada sistema de información, de su función, de las posibilidades económicas de la organización y de cuáles sean los riesgos a los que esté expuesto el sistema.

2.4.4.Seguridad lógica

Los mecanismos y herramientas de seguridad lógica tienen como objetivo proteger digitalmente la información de manera directa.

2.4.4.1.Control de acceso

Se controla el acceso mediante nombres de usuarios y contraseñas.

2.4.4.2.Cifrado de datos (encriptación)

Los datos se enmascaran con una clave especial creada mediante un algoritmo de encriptación. Emisor y receptor son conocedores de la clave y a la llegada del mensaje se produce el descifrado. El cifrado de datos fortalece la confidencialidad.

2.4.4.3.Antivirus

Detectan e impiden la entrada de virus y otras formas de *software* malicioso. En el caso de infección tienen la capacidad de eliminarlos y de corregir los daños que ocasionan en el sistema.

2.4.4.4.Cortafuegos (firewall)

Se trata de uno o más dispositivos de *software*, de *hardware* o mixtos que permiten, deniegan o restringen el acceso al sistema. Protege la integridad de la información.

2.4.4.5.Firma digital

Se trata de uno o más dispositivos de *software* o de *hardware* utilizados para la transmisión de mensajes telemáticos y en la gestión de documentos electrónicos (por ejemplo, gestiones en oficinas virtuales). Su finalidad es identificar en forma segura a la persona o al equipo que se hace responsable del mensaje o del documento. Protege la integridad y la confidencialidad de la información.

2.4.4.6.Certificados digitales

Son documentos digitales que le permiten a una entidad autorizada garantizar que una persona es quien dice ser, avalada por la verificación de su clave pública. Protege la integridad y la confidencialidad de la información.

Las redes inalámbricas (*WiFi*) necesitan precauciones adicionales para su protección:

2.4.4.6.1.Usar un SSID

El SSID (Services Set Identifier) quiere decir, darle un nombre a la red, preferiblemente uno que no llame la atención de terceros que detecten esta red entre las disponibles. Se debe cambiar con cierta frecuencia el SSID.

2.4.4.6.2.Protección de la red mediante claves encriptadas WEP o WPA

WEP: Wired Equivalent Privacy o WAP: WiFi Protected Access. La clave *WEP* consume más recursos y es más fácilmente descifrable que la *WPA* y debería cambiarse con frecuencia. La *WPA* es de encriptación dinámica y mucho más segura al ser más difícil de descifrar. Es recomendable cambiar periódicamente la contraseña de acceso a la red.

2.4.4.6.3.Filtrado de direcciones MAC

MAC: Media Access Control. Es un mecanismo de acceso al sistema mediante *hardware*, por el que se admiten solo determinadas direcciones, teniendo en cuenta que cada tarjeta de red tiene una dirección MAC única en el mundo. Puede resultar engorroso de configurar y no es infalible puesto que es posible disfrazar la dirección *MAC* real.

2.4.5.Enfoque global de la seguridad

La información es el núcleo de todo sistema de información. Para proteger sus propiedades de integridad, disponibilidad y confidencialidad es necesario tener en cuenta los niveles que la rodean para dotarlos de mecanismos y servicios de seguridad.

Desde el exterior hasta llegar la información se pueden definir estos niveles:

- Edificio y habitaciones: Edificio, planta o habitaciones, por ser el lugar físico donde se encuentran ubicados los demás niveles.

- *Hardware* y los componentes de la red que se encuentran en el interior del entorno físico, porque contienen, soportan y distribuyen la información.

- Sistema operativo y todo el *software*, porque gestiona la información.

- Conexión a internet, por ser la vía de contacto entre el sistema de información y el exterior.

- La información.

En el edificio habrá antenas, cableado en los muros, etc. Entre el *hardware* contamos con *routers*, *switches*, ordenadores, servidores, periféricos, etc. El sistema operativo y el *software* gestionan los accesos a internet. La información es el bien preciado que no se debe descuidar, pues desde internet solamente se podrá acceder a una parte de ella siempre que los usuarios tengan autorización.

2.4.6.Legislación sobre seguridad informática y protección de datos personales.

El delito informático implica actividades criminales que los países han tratado de encuadrar en figuras típicas de carácter tradicional, tales como robos, hurtos, fraudes, falsificaciones, perjuicios, estafas, sabotajes. Sin embargo, debe destacarse que el uso de las técnicas informáticas ha creado nuevas posibilidades del uso indebido de las computadoras, lo que ha creado la necesidad de regulación por parte del derecho.

Se considera que no existe una definición formal y universal de delito informático pero se han formulado conceptos respondiendo a realidades nacionales concretas: "no es labor fácil dar un concepto sobre delitos informáticos, en razón de que su misma denominación alude a una situación muy especial, ya que para hablar de "delitos" en el sentido de acciones típicas, es decir tipificadas o contempladas en textos jurídicos penales, se requiere que la expresión "delitos informáticos" esté consignada en los códigos penales, lo cual en nuestro país, al igual que en otros muchos no han sido objeto de tipificación aún." [2]

[2] *TÉLLES VALDEZ, Julio. Derecho Informático. 2° Edición. Mc Graw Hill. México. 1996 Pág. 103-104*

En 1983, la Organización para la Cooperación y Desarrollo Económicos (OCDE) inició un estudio de las posibilidades de aplicar y armonizar en el plano internacional las leyes penales a fin de luchar contra el problema del uso indebido de los programas computacionales.

En 1992 la Asociación Internacional de Derecho Penal, durante el coloquio celebrado en Wurzburgo (Alemania), adoptó diversas recomendaciones respecto a los delitos informáticos, entre ellas que, en la medida que el Derecho Penal no sea suficiente, deberá promoverse la modificación de la definición de los delitos existentes o la creación de otros nuevos, si no basta con la adopción de otras medidas como por ejemplo el "principio de subsidiariedad"[3].

Se entiende Delito como: "acción penada por las leyes por realizarse en perjuicio de algo o alguien, o por ser contraria a lo establecido por aquéllas".[4]

Finalmente la OCDE publicó un estudio sobre delitos informáticos y el análisis de la normativa jurídica en donde se reseñan las normas legislativas vigentes y se define Delito Informático como "cualquier comportamiento antijurídico, no ético o no autorizado, relacionado con el procesado automático de datos y/o transmisiones de datos."[5]

"Los delitos informáticos se realizan necesariamente con la ayuda de los sistemas informáticos, pero tienen como objeto del injusto la información en sí misma".[6]

Adicionalmente, la OCDE elaboró un conjunto de normas para la seguridad de los sistemas de información, con la intención de ofrecer las bases para que los distintos países pudieran erigir un marco de seguridad para los sistemas informáticos.

- En esta delincuencia se trata con especialistas capaces de efectuar el crimen y borrar toda huella de los hechos, resultando, muchas veces, imposible de deducir cómo es y cómo se realizó el delito. La Informática reúne características que la convierten en un medio idóneo para la comisión de nuevos tipos de delitos que en gran parte del mundo ni siquiera han podido ser catalogados.

- La legislación sobre sistemas informáticos debería perseguir acercarse lo más posible a los distintos medios de protección ya existentes, pero creando una nueva regulación basada en los aspectos del objeto a proteger: la información.

---

[3] *El principio de subsidiariedad, dispone que un asunto debe ser resuelto por la autoridad (normativa, política o económica) más próxima al objeto del problema. Es uno de los principios sobre los que se sustenta la Unión Europea, según quedó establecido por el Tratado de Maastricht, firmado el 7 de febrero de 1992 y después conocido como Tratado de la Unión Europea. Su actual formulación quedó plasmada en el Artículo 5 (2), modificada por el Tratado de Lisboa desde el 1º de diciembre de 2009. Un análisis descriptivo de este principio puede encontrarse en el Protocolo 30 sobre la aplicación de los principios de subsidiariedad y proporcionalidad, anejo al Tratado. Fuente: http://es.wikipedia.org/wiki/Principio_de_subsidiariedad*

[4] *MOLINER, María. Diccionario de María Moliner Edición Digital. Copyright© 1996 Novel Inc.; Copyright © 1996 Maria Moliner.*

[5] Definición elaborada por un Grupo de Expertos, invitados por la OCDE a París en Mayo de 1993.

[6] CARRION, Hugo Daniel. Tesis "Presupuestos para la Punibilidad del Hacking". Julio 2001. http://www.delitosinformaticos.com/tesis.htm

En este punto debe hacerse un punto y notar lo siguiente:

- No es la computadora la que atenta contra el hombre, es el hombre el que encontró una nueva herramienta, quizás la más poderosa hasta el momento, para delinquir.

- No es la computadora la que afecta nuestra vida privada, sino el aprovechamiento que hacen ciertos individuos de los datos que ellas contienen.

- La humanidad no está frente al peligro de la informática sino frente a individuos sin escrúpulos con aspiraciones de obtener el poder que significa el conocimiento.

- Por eso la amenaza futura será directamente proporcional a los adelantos de las tecnologías informáticas.

- La protección de los sistemas informáticos puede abordarse desde distintos perspectivas: civil, penal, comercial o administrativa.

Lo que se deberá intentar es que ninguna de ellas sea excluyente con las demás y, todo lo contrario, lograr una protección global desde los distintos sectores para alcanzar cierta eficiencia en la defensa de estos sistemas informáticos.

Julio Téllez Valdez clasifica a los delitos informáticos en base a dos criterios:

- Como instrumento o medio: se tienen a las conductas criminales que se valen de las computadoras como método, medio o símbolo en la comisión del ilícito.

Ejemplos:

- Falsificación de documentos vía computarizada: tarjetas de créditos, cheques, etc.

- Variación de la situación contable

- Planeamiento y simulación de delitos convencionales como robo, homicidio y fraude

- Alteración del funcionamiento normal de un sistema mediante la introducción de código extraño al mismo: virus, bombas lógicas, etc.

- Intervención de líneas de comunicación de datos o teleprocesos

- Como fin u objetivo: se enmarcan las conductas criminales que van dirigidas en contra de la computadora, accesorios o programas como entidad física

Ejemplos:

- Instrucciones que producen un bloqueo parcial o total del sistema.

- Destrucción de programas por cualquier método.

- Atentado físico contra la computadora, sus accesorios o sus medios de comunicación.

- Secuestro de soportes magnéticos con información valiosa, para ser utilizada con fines delictivos.

Este mismo autor sostiene que las acciones delictivas informáticas presentan las siguientes características:

- Sólo una determinada cantidad de personas (con conocimientos técnicos por encima de lo normal) pueden llegar a cometerlos.

- Son conductas criminales del tipo "cuello blanco": no de acuerdo al interés protegido (como en los delitos convencionales) sino de acuerdo al sujeto que los comete. Generalmente este sujeto tiene cierto status socioeconómico y la comisión del delito no puede explicarse por pobreza, carencia de recursos, baja educación, poca inteligencia, ni por inestabilidad emocional.

- Son acciones ocupacionales, ya que generalmente se realizan cuando el sujeto atacado se encuentra trabajando.

- Son acciones de oportunidad, ya que se aprovecha una ocasión creada por el atacante.

- Provocan pérdidas económicas.

- Ofrecen posibilidades de tiempo y espacio.

- Son muchos los casos y pocas las denuncias, y todo ello por la falta de regulación y por miedo al descrédito de la organización atacada.

- Presentan grandes dificultades para su comprobación, por su carácter técnico.

- Tienden a proliferar, por lo se requiere su urgente regulación legal.

María Luz Lima, por su parte, presenta la siguiente clasificación de "delitos electrónicos"[7]:

- Como Método: conductas criminales en donde los individuos utilizan métodos electrónicos para llegar a un resultado ilícito.

- Como Medio: conductas criminales en donde para realizar un delito utilizan una computadora como medio o símbolo.

- Como Fin: conductas criminales dirigidas contra la entidad física del objeto o máquina electrónica o su material con objeto de dañarla.

2.4.6.1.Datos Personales y Privacidad

CONSTITUCIÓN DE LA NACIÓN ARGENTINA, modificada en 1994. Art. 19 y 43(Habeas Data).

LEY 25.326 de Protección de los Datos Personales.

DECRETO 1.558/2001, que reglamenta la Ley de Protección de los Datos Personales

LEY 24.766 de Confidencialidad sobre Información y Productos que estén legítimamente bajo control de una persona y se divulgue indebidamente de manera contraria a los usos comerciales honestos.

[7] *LIMA de la LUZ, María. Criminalia N° 1-6 Año L. Delitos Electrónicos. Ediciones Porrua. México. Enero-Julio 1984.*

LEY 26.529 Derechos del Paciente en su Relación con los Profesionales e Instituciones de la Salud, que regula los derechos del paciente sobre su historia clínica y el consentimiento informado luego de recibir la información sobre su tratamiento.

LEY 23.592 de Actos Discriminatorios, para quienes realicen por cualquier medio actos discriminatorios determinados por motivos de raza, religión, nacionalidad, ideología, opinión política o gremial, sexo, posición económica, condición social o caracteres físicos.

LEY 23.511 Banco Nacional de Datos Genéticos (BNDG), que crea el BNDG a fin de obtener, almacenar y analizar información genética que facilite el esclarecimiento de conflictos de filiación y para esclarecer delitos de lesa humanidad mediante la identificación de personas desaparecidas (adicionado por la Ley 26.548).

DECRETO 38/2013, que reglamenta la Ley 26.548 sobre el Registro Nacional de Datos Genéticos (BNDG).

CÓDIGO CONTRAVENCIONAL DE LA CIUDAD DE BUENOS AIRES, en su art. 52 castiga con multa o arresto al que intimide u hostigue a otro de modo amenazante.

LEY 2602 de la Ciudad Autónoma de Buenos Aires, que regula la utilización por parte del Poder Ejecutivo de videocámaras para grabar imágenes en lugares públicos y su posterior tratamiento.

LEY 3130 de la Ciudad Autónoma de Buenos Aires, que modifica la Ley 2602 y dispone que las cámaras deberán tener un dispositivo de emergencia que recibirá el Centro Único de Comando y Control (CUCC).

DECRETO 716/09 de la Ciudad Autónoma de Buenos Aires, que regula la utilización por parte del Poder Ejecutivo de videocámaras para grabar imágenes en lugares públicos y aprueba reglamentación de la Ley 2602.

DECRETO 1119/09 de la Ciudad Autónoma de Buenos Aires, que modifica Decreto 716/09 sobre la Utilización por parte del Poder Ejecutivo de videocámaras para grabar imágenes en lugares públicos.

LEY 21.173, incorpora al Código Civil el art. 1071 bis, que sanciona al que arbitrariamente se entrometa en la vida ajena, publique retratos, difunda correspondencia, mortifique a otros en sus costumbres o sentimientos, o perturbe de cualquier modo su intimidad.

2.4.6.2.Disposiciones de la Dirección Nacional de Protección de Datos Personales (DNDP)

DISPOSICIÓN Nº 2/2003, de la Dirección Nacional de Protección de Datos Personales (DNPDP), habilita el Registro Nacional de Bases de Datos y dispone la realización del Primer Censo Nacional de Bases de Datos.

DISPOSICION Nº 1/2004, de la Dirección Nacional de Protección de Datos Personales (DNPDP), se implementa, con carácter obligatorio, el Primer Censo Nacional de Archivos, Registros, Bases o Bancos de Datos Privados.

DISPOSICION Nº 4/2004, de la Dirección Nacional de Protección de Datos Personales (DNPDP), se homologa el Código de Ética de la Asociación de Marketing Directo e Interactivo de Argentina (AMDIA).

DISPOSICION Nº 2/2005, de la Dirección Nacional de Protección de Datos Personales (DNPDP), implementa el Registro Nacional de Bases de Datos y los formularios de inscripción.

DISPOSICION Nº 7/2005, de la Dirección Nacional de Protección de Datos Personales (DNPDP), aprueba la “Clasificación de Infracciones” y la “Graduación de las Sanciones” a aplicar ante violaciones a las normas de la Ley 25.326 y sus reglamentaciones.

DISPOSICION Nº 2/2006, de la Dirección Nacional de Protección de Datos Personales (DNPDP), implementa el Relevamiento Integral de Bases de Datos Personales del Estado Nacional.

DISPOSICION Nº 11/2006, de la Dirección Nacional de Protección de Datos Personales (DNPDP), se aprueban las “Medidas de Seguridad para el Tratamiento y Conservación de los Datos Personales Contenidos en Archivos, Registros, Bancos y Bases de Datos Públicos no estatales y Privados”.

DISPOSICION Nº 2/2008, de la Dirección Nacional de Protección de Datos Personales (DNPDP), se crea el Repertorio de Jurisprudencia sobre Hábeas Data en el ámbito de la DNPDP y de libre consulta.

DISPOSICION Nº 3/2008, de la Dirección Nacional de Protección de Datos Personales (DNPDP), se crea el Centro de Jurisprudencia, Investigación y Promoción de la Protección de los Datos Personales en el ámbito de la DNPDP.

DISPOSICION Nº 3/2012, de la Dirección Nacional de Protección de Datos Personales (DNPDP), aprueba las Normas de Inspección e Instructivo del Formulario de Inspección de la DNPDP y deroga la Disposición DNPDP N° 05/2008.

DISPOSICION Nº 6/2008, de la Dirección Nacional de Protección de Datos Personales (DNPDP), aprueba el Procedimiento de Control en la Ejecución de Formularios de Consentimiento Informado en ensayos de farmacología clínica.

DISPOSICION Nº 7/2008, de la Dirección Nacional de Protección de Datos Personales (DNPDP), aprueba la “Guía de Buenas Prácticas en Políticas de Privacidad para las Bases de Datos del Ámbito Público” y el texto modelo de “Convenio de Confidencialidad”.

DISPOSICION Nº 10/2008, de la Dirección Nacional de Protección de Datos Personales (DNPDP), establece que los responsables y usuarios de bancos de datos públicos o privados, deberán incluir información específica legal en su página web y en toda comunicación o publicidad, y en los formularios utilizados para la recolección de datos.

DISPOSICION Nº 4/2009, de la Dirección Nacional de Protección de Datos Personales (DNPDP), establece que la opción para el ejercicio del derecho de retiro o bloqueo contemplada en el artículo 27, inciso 3, de la Ley 25.326, deberá aparecer en toda comunicación que se efectúe con fines publicitarios, junto con el mecanismo previsto para su ejercicio.

DISPOSICION Nº 7/2010, de la Dirección Nacional de Protección de Datos Personales (DNPDP), se crea el Centro de Asistencia a las Víctimas de Robo de Identidad en el Ámbito de la Dirección Nacional de Protección de Datos Personales.

DISPOSICION Nº 12/2010, de la Dirección Nacional de Protección de Datos Personales (DNPDP), establece que al tratarse datos destinados a difusión pública que contengan datos

sensibles o referente a menores, incapaces y asuntos de familia deberán aplicarse procedimientos de disociación y de protección a fin de evitar la identificación del titular del dato.

DISPOSICION Nº 17/2010, de la Dirección Nacional de Protección de Datos Personales (DNPDP), se establece el sistema informativo denominado “Base Informática para la Comunicación Electrónica Interjurisdiccional sobre Datos Personales en Información Crediticia”.

DISPOSICION Nº 24/2010, de la Dirección Nacional de Protección de Datos Personales (DNPDP), se crea el Registro Nacional de Documentos de Identidad Cuestionados.

DISPOSICION Nº 3/2012, de la Dirección Nacional de Protección de Datos Personales (DNPDP), aprueba el “Formulario de Inspección” y el “Instructivo del Formulario de Inspección”.

DISPOSICION Nº 4/2012, de la Dirección Nacional de Protección de Datos Personales (DNPDP), sustituye el art. 7° de la Disposición DNPDP Nº 02/05 estableciendo que no será necesaria la renovación anual cuando la cantidad de personas en el total de las bases de datos sea menor a 5.000, y se declare que no se realiza tratamiento de datos sensibles.

2.4.6.3.Delitos Informáticos y Ciberseguridad

CÓDIGO PENAL DE LA NACIÓN ARGENTINA.

LEY 26.388 de Ley de Delitos Informáticos.

Convención de Budapest sobre Ciberdelincuencia.

Declaración “Fortalecimiento de la Seguridad Cibernética en las Américas”. Comité Interamericano Contra el Terrorismo (CICTE) (OEA) (aprobado durante la cuarta sesión plenaria, el 7 de marzo de 2012)

Declaración de Panamá sobre “La Protección de la Infraestructura Crítica en el Hemisferio frente al Terrorismo”. Comité Interamericano Contra el Terrorismo (CICTE) (OEA) (aprobada en la tercera sesión plenaria, el 1 de marzo de 2007)

LEY 2.257 del Gobierno de la Ciudad de Buenos Aires, aprueba el Convenio N° 14/04, “Convenio de Transferencia Progresiva de Competencias Penales de la Justicia Nacional al Poder Judicial de la Ciudad Autónoma de Buenos Aires”, suscripto entre el Gobierno Nacional y el Gobierno de la Ciudad Autónoma de Buenos Aires, como ser pornografía infantil, exhibiciones obscenas, amenazas y daños informáticos, ente otros.

RESOLUCIÓN 501/FG/12 de la Fiscalía General de la Ciudad Autónoma de Buenos Aires, aprueba en calidad de prueba piloto la implementación del Equipo Fiscal “A” de la Unidad Fiscal Este especializado en delitos y contravenciones informáticas, que actuará con competencia especial única en toda la Ciudad Autónoma de Buenos Aires.

RESOLUCIÓN 580/2011 de la Jefatura de Gabinete de Ministros, crea el Programa Nacional de Infraestructuras Críticas de Información y Ciberseguridad en el ámbito de la Oficina Nacional de Tecnologías de Información (ONTI).

DECRETO 1766/2011, crea el Sistema Federal de Identificación Biométrica para la Seguridad (SIBIOS) que tendrá por objeto prestar un servicio centralizado de información respecto de los registros patronímicos y biológicos individuales, a los fines de contribuir a la comprobación idónea y oportuna en materia de identificación de personas y rastros, en procura de optimizar la investigación científica de delitos y el apoyo a la función preventiva de seguridad.

DECISIÓN ADMINISTRATIVA 669/2004 de la Jefatura de Gabinete de Ministros, que establece que los organismos del Sector Público Nacional deberán dictar o adecuar sus políticas de seguridad y conformar Comités de Seguridad en la Información.

DISPOSICIÓN 6/2005 de la Oficina Nacional de Tecnologías de Información (ONTI), que aprueba la “Política de Seguridad de la Información Modelo” para el Sector Público Nacional.

LEY 863 de la Legislatura de la Ciudad Autónoma de Buenos Aires, establece que los establecimientos comerciales que brinden acceso a Internet deben instalar y activar filtros de contenido sobre páginas pornográficas.

CÓDIGO CONTRAVENCIONAL DE LA CIUDAD DE BUENOS AIRES, en su art. 61 castiga al que tolere o admita la presencia de menores en lugares no autorizados (local de espectáculos públicos, de baile o de entretenimientos tipo ciber) y en su art. 62 castiga al que suministre o permita a un menor el acceso a material pornográfico.

COMUNICACIÓN “B” 9042 del Banco Central de la República Argentina (BCRA), relativa a los requisitos mínimos de gestión, implementación y control de los riesgos relacionados con Tecnología Informática y Sistemas de Información y recursos asociados para las entidades financieras.

2.4.6.4.Comercio Electrónico y contratación electrónica

RESOLUCIÓN 412/99 del Ministerio de Economía y Obras y Servicios Públicos. Recomendaciones del Grupo de Trabajo sobre Comercio Electrónico y Comercio Exterior.

RESOLUCIÓN 104/2005 de la Secretaría de Coordinación Técnica, que incorpora al ordenamiento jurídico nacional la Resolución Nº 21 del MERCOSUR, relativa al Derecho de Información al Consumidor en las Transacciones Comerciales Efectuadas por Internet.

DECRETO 1023/2001, sobre el Régimen de Contrataciones de la Administración Pública Nacional, donde establece en Capítulo II las Contrataciones Públicas Electrónicas.

LEY 2.244 de la Ciudad Autónoma de Buenos Aires, establece que las personas que comercialicen o presten servicios a consumidores y/o usuarios en el ámbito de la C.A.B.A. y posean página web, deberán agregar un enlace con la Dirección General de Defensa y Protección al Consumidor.

LEY 2.817 de la Ciudad Autónoma de Buenos Aires, fija obligaciones a los Proveedores de Bienes o Servicios con respecto a los Consumidores.

RESOLUCIÓN 412/99 del Ministerio de Economía, Obras y Servicios Públicos, aprueba Recomendaciones formuladas por el Grupo de Trabajo sobre Comercio Electrónico y Comercio Exterior del Ministerio.

LEY 26.104, Publicidad con Fines Turísticos, establece que toda publicidad contenida en medios electrónicos, cuyas imágenes exhiban atractivos turísticos, deberá indicar cierta información específica.

LEY 24.240 de Defensa al Consumidor.

RESOLUCIÓN 33.463/08 de la Superintendencia de Seguros de la Nación, que incorpora al Reglamento de la Actividad Aseguradora la entrega de documentación por medios electrónicos.

RESOLUCIÓN 7/2002 de la Secretaría de la Competencia, la Desregulación y la Defensa del Consumidor, que establece los mecanismos que garantizan el derecho de los consumidores a recibir la más completa información acerca de los precios de los bienes y servicios que les son ofrecidos.

RESOLUCIÓN 53/2003 de la Secretaría de la Competencia, la Desregulación y la Defensa del Consumidor, que determina las cláusulas que no podrán ser incluidas en los contratos de consumo, por ser opuestas a los criterios establecidos en el art. 37 de la Ley N° 24.240 y su reglamentación.

RESOLUCIÓN 26/2003 de la Secretaría de Coordinación Técnica, que deroga la Disposición 3/2003 de la Subsecretaría de Defensa de la Competencia y Defensa del Consumidor y modifica la Resolución de la ex Secretaría de la Competencia, la Desregulación y la Defensa del Consumidor 53/2003, prorrogando el plazo del art. 2º de la misma y definiendo las cláusulas consideradas abusivas en los contratos suscriptos por los consumidores y usuarios de bienes y servicios.

2.4.6.5.Documento Electrónico

LEY 24.624 de Presupuesto General de la Administración Nacional (1996), modificatoria de la Ley 11.672, que considera con pleno valor probatorio a la documentación de la Administración Pública Nacional archivada en soportes electrónicos.

LEY 26.685 de Expediente Electrónico, que autoriza el uso del expediente electrónico, documentos electrónicos, firma digital, comunicaciones electrónicas y domicilio electrónico constituido, en todos los procesos ante el Poder Judicial de la Nación.

2.4.6.6.Firma Digital y Electrónica

LEY 25.506 de Firma Digital.

DECRETO 2.628/02 que reglamenta la Ley Nº 25.506 de Firma Digital.

RESOLUCIÓN 45/97 de la Secretaría de la Función Pública que Incorpora la tecnología de Firma Digital a los procesos de información del sector público.

RESOLUCIÓN 194/98 de la Secretaría de la Función Pública, que aprueba los estándares aplicables a la “Infraestructura de Firma Digital para el Sector Público Nacional”.

DECRETO 427/98, implementa el régimen para el empleo de la Firma Digital en actos internos de la Administración Pública Nacional con los mismos efectos de la firma ológrafa.

DECRETO 283/2003, autoriza a la Oficina Nacional de Tecnologías Informáticas (ONTI) a proveer certificados digitales para utilizarse en los circuitos de la Administración Pública Nacional que requieran Firma Digital.

DECRETO 1028/2003, disuelve el Ente Administrador de Firma Digital creado por el Decreto 2628/2002 y lo reemplaza por la Oficina Nacional de Tecnologías de Información (ONTI) de la Subsecretaría de la Gestión Pública.

DECISIÓN ADMINISTRATIVA 6/2007 de la Jefatura de Gabinete de Ministros, establece el marco normativo de Firma Digital aplicable al otorgamiento y revocación de las licencias a los certificadores que así lo soliciten.

2.4.7.Encriptación

2.4.7.1.Criptografía

Según el diccionario de la Real Academia Española, la palabra criptografía proviene de la unión de los términos griegos κρύπτω krypto (oculto), y γράφως graphos (escribir), literalmente «escritura oculta» de la información. Entre las disciplinas que englobaba cabe destacar la Teoría de la Información, la Teoría de Números o Matemática Discreta, que estudia las propiedades de los números enteros, y la complejidad Algorítmica. Existen dos trabajos fundamentales sobre los que se apoya prácticamente toda la teoría criptográfica actual. Uno de ellos, desarrollado por Claude Shannon en sus artículos “A Mathematical Theory of Comunication” (1948) y “Communication Theory of Secrecy System” (1949), sienta las bases de la teoría de la información y de la criptografía moderna. El segundo publicado Whitfiel Diffie y Matin Hellman en 1976, se titulaba “New direction in Cryptography” e introducía el concepto de Criptografía Asimétrica, abriendo enormemente el abanico de aplicación de esta disciplina.

Conviene hacer notar que la palabra Criptografía solo hace referencia al uso de códigos, conocidas en su conjunto como Criptoanálisis. En cualquier caso ambas disciplinas están íntimamente ligadas; no olvidemos que cuando se diseña un sistema para cifrar información, hay que tener muy presente su posible criptoanálisis, ya que en caso contrario podríamos llevarnos desagradables sorpresas.

2.4.7.2.Seguridad

El concepto de seguridad en la información es mucho más amplio que la simple protección de los datos a nivel lógico. Para proporcionar seguridad real hemos de tener en cuenta múltiples

factores, tanto internos como externos. En primer lugar habría que caracterizar el sistema que va a albergar la información para poder identificar las amenazas, y en este sentido podríamos hacer la siguiente subdivisión:

- Sistemas aislados. Son los que no están conectados a ningún tupo de red. De unos años a esta parte se han convertido en minoría, debido al auge que ha experimentado internet.

- Sistemas interconectados. Hoy por hoy casi cualquier ordenador pertenece a alguna red, enviando y recogiendo información del exterior casi constantemente. Esto hace que las redes de ordenadores sean cada día más complejas y supongan un peligro potencial que no se puede, en ningún caso, ser ignorado.

En cuanto a las cuestiones que hemos de fijar podríamos clasificarlas de la siguiente forma

- Seguridad física: Englobaremos dentro de esta categoría a todos los asuntos relacionados con la salvaguarda de los soportes físicos de la información propiamente dicha. En este nivel estaría, entre otras, las medidas contra incendios y sobrecargas eléctricas, la prevención de ataques terroristas, las políticas de copias de respaldo, etc. También se suelen tener en cuenta dentro de este punto aspectos relacionados con la restricción de acceso físico a las computadoras únicamente a personas autorizadas.

- Seguridad de la información: En este apartado prestaremos atención a la preservación de la información frente a observadores no autorizados. Para ello podemos emplear tanto criptografía simétrica como asimétrica, estando la primera únicamente indicada en sistemas aislados, ya que si la empleáramos en redes, al tener que transmitir la clave por el canal de comunicación, estaríamos asumiendo un riesgo excesivo.

- Seguridad del canal de comunicación: Los canales de comunicación rara vez se consideran seguros. Debido a que en la mayoría de los casos escapan a nuestro control, ya que pertenecen a terceros, resulta imposible asegurarse totalmente de que no están siendo escuchados o intervenidos.

- Problemas de autentificación: Debido a los problemas del canal de comunicación, es necesario asegurarse de que la información que recibimos en la computadora viene de quien creemos que viene y que además no ha sido alterada. Para esto se suele emplear criptografía asimétrica en conjunción con funciones resumen (hash).

- Problemas de suplantación: en las redes tenemos el problema añadido de que cualquier usuario autorizado puede acceder al sistema desde fuera, por lo que hemos de confiar en sistemas fiables para garantizar que los usuarios no están siendo suplantados por intrusos. Para conseguir esto normalmente se emplean mecanismos basados en contraseñas.

- No repudio: cuando se recibe un mensaje no solo es necesario poder identificar de forma unívoca al remitente, sino también es necesario que asuma todas las responsabilidades derivadas de la información que haya podido enviar. En este sentido es fundamental impedir que el emisor pueda repudiar un mensaje, es decir, negar su autoría sobre él.

- Anonimato: es, en cierta manera, el concepto opuesto al del no repudio. En determinadas aplicaciones, como puede ser un proceso electoral, o la simple de violaciones de los derechos humanos en entornos dictatoriales, es crucial garantizar el anonimato del ciudadano para poder preservar su intimidad y su libertad. Es una característica realmente difícil de conseguir y, desafortunadamente, no goza de muy buena fama, especialmente en países donde prima la seguridad nacional sobre la libertad y la intimidad de los ciudadanos.

2.4.7.3.Autentificación

En general, y debido a los diferentes tipos de situaciones que podemos encontrar en un sistema informático, distinguiremos tres tipos de autentificación:

- Autentificación de mensaje: Queremos garantizar la procedencia de un mensaje conocido, de forma que podamos asegurarnos de que no es una falsificación. Este mecanismo se conoce habitualmente como firma digital.

- Autentificación de usuario mediante contraseña: En este caso se trata de garantizar la presencia de un usuario legal en el sistema. El usuario deberá poseer una contraseña secreta que le permita identificarse.

- Autentificación de dispositivo: Se trata de garantizar la presencia frente al sistema de un dispositivo concreto. Este dispositivo puede estar solo o tratarse de una llave electrónica que

sustituye o complementa a la contraseña para identificar a un usuario.

Nótese que la autentificación de usuario por medio de alguna característica biométrica, como pueden ser las huellas digitales, la retina, el iris, la voz, etc. puede reducirse a un problema de autentificación de dispositivo, solo que el dispositivo en este caso es el propio usuario.

2.4.7.4.Protocolos de Criptografía

Los siguientes temas son resumidos del capítulo 2 del libro de Criptografía[8]:

2.4.7.5.Criptografía Simétrica

Las dos partes quedan involucradas, una llamada “A” que pretende enviar un mensaje a otra “B”. Estas dos partes como primer paso, acordarán el uso de una forma de criptografía o cripto-sistema (protocolo, algoritmo) Luego acordarán una llave o clave. La parte “A” encriptará el mensaje (texto-plano), valiéndose del algoritmo y de la llave. El resultado de esta última operación será la obtención del texto-cifrado correspondiente a la función del algoritmo, con el texto-plano y la llave como entradas. La parte “A” enviará a “B” el texto cifrado. La parte “B”, entonces, utilizando el mismo algoritmo y la llave acordada previamente, podrá desencriptar el texto-cifrado para obtener el texto-plano, es decir, el mensaje original.

Por supuesto, el medio por donde se enviará el texto-cifrado se asume inseguro. Hemos de seleccionar un algoritmo seguro, estandarizado y suficientemente probado por la comunidad académica – y la industria y entes gubernamentales -, para lograr contar con la seguridad de que, solo a partir del texto-cifrado, una tercera parte no podrá desencriptar el mensaje.

Lo que la parte “A” y “B” deben acordar en secreto, previamente a comunicarse utilizando criptografía simétrica, es la llave que han de resguardar con el mayor de los cuidados. No sucede

[8] Maiorano, A. (2009). *Criptografia, Técinas de desarrollo para profesionales.* México D.F.: ALFAOMEGA GRUPO EDITOR, S.A.

igual con la elección del algoritmo; no debe importarnos que la tercera persona conozca o no el algoritmo utilizado para la encriptación, ya que eso podría acordarse públicamente. Pero, al hablar de criptografía simétrica, utilizando un algoritmo comprobado, la seguridad de las comunicaciones estará basada en mantener en secreto la llave entre las partes que sí deben acceder a la información cifrada.

Esto último representa quizá el problema o aspecto negativo más importante, característico de la criptografía simétrica. Las llaves deben acordarse (o distribuirse) en secreto. Ha de ser tan importante luego su resguardo como la información que se ha de cifrar, ya que, conocida la llave, se podrá obtener el mensaje.

Con referencia a los potenciales problemas para considerar a la hora de implementar esta metodología, particularmente para la comunicación segura entre más partes, será prudente tener en cuenta que por cada par de usuarios, o partes, se necesitará una llave. Si usted formase parte de un equipo de cuatro personas, deberá disponer de tres llaves, una para comunicarse con cada una de las partes y lo mismo disponer de tres llaves, una para comunicarse con cada una de las partes, y lo mismo corre para el resto. De esta manera, entre las cuatro partes se manejarán en todo el grupo: Tres llaves (las propias) más otras tres llaves (compartidas entre sí por las tres partes), haciendo un total de seis llaves. La fórmula para calcular la cantidad de llaves es n(n-1)/2, siendo n la cantidad de partes. Como se desprende de la fórmula al incrementarse la cantidad de partes crecerá rápidamente la cantidad de llaves para manejar. Al pensar en un grupo o equipo numeroso, como un conjunto que se ha de administrar, donde las partes o usuarios deben ocuparse del resguardo de tantas llaves, la opción más conveniente para recomendar es la utilización de criptografía asimétrica de una vía y hash.

2.4.7.6.Funciones de una vía

El concepto general de las funciones de una vía es que, a través de ellas, se podrá computar su resultado de manera relativamente rápida, pero en cambio la obtención de entrada (el parámetro de la función) a partir del resultado será prácticamente inviable. Esto quiere decir que siendo f la función de una vía, se podrá calcular f(x) de manera sencilla, pero obtener x demandará años,

aunque dispongamos de toda la capacidad de procesamiento que podamos adquirir.

Deberá quedar en claro también que una función de una vía no se trata de un protocolo criptográfico en sí. No se utilizan para encriptar información, sino que forman parte fundamental de muchos algoritmos y técnicas criptográficas. Es relevante la aclaración ya que no cifran ni descifran información.

Estrictamente, no existen comprobaciones matemáticas de la existencia de funciones con la característica de la imposibilidad de cálculo de su inversa, ni siquiera de la posibilidad de construirlas. Existen, si muchas que lo parecen, que permiten su cálculo de forma rápida y que, hasta el momento, se desconoce una forma sencilla de obtener su inversa.

2.4.7.6.1.Hash

Con algo más de especificidad encontramos, dentro de las funciones de una vía, las funciones *hash* criptográfico. Estas tienen varios sinónimos en la literatura: Resúmenes de mensajes (*message digest*), huellas digitales (*finger prints*), funciones de compresión, funciones de contracción chequeos de integridad de mensajes (*message integrity check* o *MIC*), códigos de detección de manipulación (*manipulation detection code* o *MDC*) y *checksums* criptográficos.

El concepto principal de estas funciones es que tomarán como entrada información o un mensaje de longitud variable (una contraseña o los contenidos de un archivo que empaquete el instalador de un programa), que se denominará pre-imagen, para convertirlo en información de salida de longitud fija (en el algoritmo MD5, por ejemplo, será de 128 *bits* de longitud). A esta salida se la denominará valor *hash*.

Repitiendo en parte lo anterior, estas funciones operan en una dirección; será posible calcular el *hash* a partir de una pre-imagen, pero será inviable generar una pre-imagen cuyo *hash* corresponda a un resultado particular.

En relación con esto, la cualidad de ser libre de colisiones es otra característica importante de estas funciones. Esto significa que una función de *hashing* debería hacer muy poco probable que dos pre-imágenes distintas generen un mismo *hash* o resultado.

En la actualidad, además de servir de base a otros algoritmos o protocolos criptográficos, los usos más comunes de estas funciones de *hashing* son la del registro de contraseñas en espacios de almacenamientos no confiables y de realizar verificaciones de integridad, generalmente sobre archivos. La generación de llaves para criptografía simétrica a partir de una contraseña, *passphrase* o frase-clave, es también otra aplicación importante para la cual nos valemos de estas funciones.

En la sección referente a la criptografía de llave pública, veremos que en los sistemas de cifrado asimétrico se hace uso de las funciones de una vía, del tipo particular *trapdoor* (de puerta trasera o tramposa, en tanto dispone de mecanismo o trampa secreta, que hará posible el cómputo de la inversa de la función).

2.4.7.7.Criptografía Asimétrica

Representó un cambio de paradigma muy importante en la materia, en 1976, cuando Whitfield Diffie y Martin Hellman la describieron explicando que implicaba la utilización de dos llaves: una pública y una privada.

Al describir el método paso por paso, usando de ejemplo las partes “A” y “B”, como cuando se describió la criptografía simétrica, podemos resumir el procedimiento de la siguiente manera: el primer paso corresponde al acuerdo entre las partes de la utilización de un protocolo de criptografía asimétrica. En el segundo paso, la parte “B” envía su llave pública a la parte A, que encriptará la información o mensaje, utilizando la llave pública recibida de la parte “B”. La parte “A” enviará a la parte “B” el resultado de esta encriptación, es decir, el texto cifrado. Por último, la parte “B” desencriptará el texto-cifrado (que fue encriptado con su propia llave pública) mediante su llave privada.

En la práctica el cifrado asimétrico NO se utiliza para encriptar mensajes, o lo que sería concretamente la información para comunicar, sino que se implementa para encriptar llaves.

Por otra parte, la criptografía asimétrica es considerablemente más lenta que la simétrica, en un orden aproximado de mil veces. Además, es vulnerable a ataques de *chosen-plaintext* o texto-plano elegido; esto quiere decir que si sabemos que el mensaje para transmitir será uno de un

conjunto de n posibles mensajes, al ser la llave del receptor pública, una tercera parte podría encriptar los n posibles mensajes o textos-planos y comparar el resultado con el texto-cifrado capturado en el medio inseguro y así inferir cuál ha sido el mensaje enviado.

Por lo tanto, y ahora nos resultará evidente, la utilización más común de la criptografía asimétrica es para hacer llegar a la otra parte una llave criptográfica simétrica y realizar así, la transmisión del mensaje o información usando criptografía simétrica en lugar de asimétrica.

2.4.8.Firma digital

Corresponde a una versión informática de la firma personal manuscrita o firma ológrafa. Se verá que en realidad, los protocolos de firma digital proporcionan, en principio, mayor seguridad que las firmas tradicionales. Estas últimas se han utilizado ampliamente como prueba de autoría o acuerdo de una parte, entre otros usos, siempre en referencia a un documento, o mejor dicho, a los contenidos de un documento. Sin embargo, como hemos mencionado pueden resultar inseguras en tanto que pueden ser, con relativa facilidad, aplicadas de manera deshonesta.

La firma digital pretende resolver estos problemas en documentos digitales. De acuerdo con la legislación de cada país, tendrá tanto carácter o validez jurídica como la firma personal manuscrita.

Respecto al protocolo la primera posibilidad para comentar será la que corresponde a la firma de documentos electrónicos, mediante criptografía simétrica y un árbitro. Aquí la parte “A” que pretende firmar un documento y enviarlo a la parte “B”, lo hará con la ayuda de una tercera parte llamada “C” -quien será el árbitro en quien se confía- de la siguiente manera: la parte “A” encriptará el mensaje que se envía a la parte “B” con su llave secreta (criptografía simétrica) pero lo envía a la parte “C”. Esta parte lo desencriptará con la llave secreta, compartida con la “A”. Al texto-plano obtenido, la parte “C” agregará una nota respecto de que ha recibido el mensaje de “A”, encriptará todo esto con la llave secreta compartida con la parte “B” –diferente de la primera, por supuesto-. Quien lo reciba, al desencriptarlo encontrará el mensaje y la certificación.

Otra posibilidad que se vale de la criptografía asimétrica es utilizar un algoritmo como RSA, que permite la encriptación y desencriptación de datos, tanto con la llave pública como la privada, la parte “A” al encriptar un mensaje con la llave privada, ya generaría un mensaje firmado. Otras

partes podrían descifrarlo con la llave pública de “A” y comprobarlo. En cambio, en otros algoritmos, como DSA (que no puede ser utilizado para encriptar y desencriptar información, ya que es un algoritmo para intercambio de llaves) se utilizan separado para la firma digital.

Otra posibilidad involucra la utilización de funciones *hash* junto con criptografía simétrica. Es la implementación más utilizada en la práctica por la ineficacia o lentitud generada la hora de firmar mensajes extensos con las metodologías anteriores. Aquí en lugar de firmar un mensaje o documento, la parte “A” firmará el *hash* resultante de tal documento. El algoritmo de *hashing* deberá acordarse de antemano. Los pasos del protocolo se resumen de la siguiente manera: la parte “A” desencriptará el *hash* que la parte “A” ha computado. Entonces, si “H” compara el *hash* que él ha generado con el que acaba de desencriptar y coinciden, podrá comprobar o verificar la firma de la parte “A”.

2.4.9.Algoritmos

A la hora de incorporar criptografía en nuestros desarrollos, nos veremos en la necesidad de optar por alguno de los algoritmos disponibles en nuestro entorno de programación o *frameworks*.

En primera instancia, deberíamos tener en claro que la encriptación asimétrica NO reemplaza a la simétrica, que ambos mecanismos proveen soluciones a problemas diferentes y, también tener en estudiado cuándo nos será conveniente la utilización de cada una. Lo mismo en tanto a las funciones de *hashing*: no debemos confundirnos pensando que pueden ser utilizadas para la encriptación y/o desencriptación de información.

Como regla general, además de las cuestiones técnicas, podemos decir que un factor para tener en cuenta debe ser el de la legalidad de la implementación y a la utilización de un algoritmo determinado en el país en donde fuera a funcionar o a distribuirse la aplicación.

Si bien convendrá tener presentes los detalles de cada uno de los algoritmos, de manera general es posible asegurar que el algoritmo DES, por ejemplo, es menos seguro que sus alternativas modernas como AES o *Twofish*. AES es el elegido en concurso para el establecimiento de un estándar federal para criptografía simétrica en los E.E. U.U., representa actualmente una porción prudente. Con respecto a los algoritmos de *hashing* SHA-1 y las diferentes versiones de

SHA – 2, son recomendadas antes que MD5, algoritmo sobre el que se han descubierto diferentes problemas recientemente, que han puesto en duda su robustez. Por último, para el caso de criptografía asimétrica, quizá contemos con menos alternativas. Lo más conservador significará la utilización del algoritmo RSA, minuciosamente analizado a través de los años, ya que en tanto se utilicen las llaves de una longitud segura, no representará, hasta donde sabemos, ningún riesgo.

2.4.9.1.DES

Data Encryption Standard, es un algoritmo de criptografía simétrica que fue aprobado por el instituto Nacional Americano de Estándares o American National Standards Institute (ANSI), en el año 1981. Antes fue aprobado también como un estándar federal del mismo país, FIPS (Federal Information Procesing Standard), en el año 1976 (publicado en 1977 como FIPS PUB 46).

Se trata de un algoritmo cifrado por bloques, que encriptará información en bloques de 64 bits de longitud. En la encriptación, un bloque de este tamaño será entrada o input del algoritmo, el texto-plano y, junto con la llave- ya que se trata de un algoritmo simétrico- se producirá una salida del mismo tamaño, el texto–cifrado.

La misma llave y el mismo algoritmo se utilizan tanto para el cifrado como para el descifrado de información (salvo por una pequeña diferencia en el manejo de la llave). La longitud de la llave es de 56 bits y es posible utilizar cualquier número que quepa en ese tamaño como llave.

2.4.9.2.Triple DES

Es un algoritmo formado a partir del algoritmo DES. Cuando los 56 bits de longitud de la llave de DES resultaron escasos, se ideó una alternativa para que, manteniendo el mismo algoritmo, se pudiese incrementar el tamaño de la llave. De esta manera, surge entonces este nuevo algoritmo, que en su versión o variante más simple no es otra cosa que la triple aplicación del algoritmo DES, con una llave que correspondería al conjunto de las tres llaves DES aplicadas.

El algoritmo opera en bloques de 64 bits de longitud de la entrada o texto-plano. Luego de una permutación inicial, el bloque es dividido en dos bloques de 32 bits. Se aplican entonces 16 rondas, de las mismas operaciones, en las cuales la información es combinada con la llave. Luego de las 16 rondas, las dos mitades citadas vuelven a juntarse para dar lugar a una última permutación.

Entonces, si B es el resultado de la iteración número *i*, L_i y R_i son las dos mitades de B_i K_i, la llave para la ronda *i* y f la función que aplica sobre la llave (permutaciones, transposiciones, operaciones XOR), una ronda implica que:

$$L_i = R_{i+1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}\ k_i)$$

2.4.9.3.AES

Advanced Encryption Standard o Estándar para la Encriptación Avanzada, es, como su predecesor DES, un algoritmo de criptografía simétrica por bloques. Es ya un estándar del gobierno de los E.E. U.U., siendo su objetivo reemplazar al algoritmo DES.

Es conocido como “Rijndael”, sobre la base de los apellidos de sus creadores Joan Daermen y Vincent Rijmen, quienes presentaron el algoritmo al Advanced Encryption Standard Contest y fue seleccionado entre otros quince algoritmos finalistas. Dado que el estándar AES ha sido algo más restrictivo, Rijndael permite un mayor rango de tamaños posibles de bloques y de longitudes de llaves. AES especifica un tamaño de bloque fijo de 128 bits de longitud; para la llave de tamaños de 128, 192 o 256 bits. Rijndael, en tanto sean múltiplo de 32 bits, las longitudes de llave y bloque podrán variar entre 128 y 256 bits.

Utiliza lo que se conoce como campo Galois (apellido de un matemático francés, precursor de esta teoría del algebra abstracta) para llevar a cabo gran parte de sus operaciones matemáticas. Este campo es una construcción matemática especial, en donde las operaciones de adición, sustracción, multiplicación y división son redefinidas y donde se dispondría de un número limitado de enteros.

Más precisamente, en Rijndael, el campo Galois utilizado solo permite un número de 8 bits (un número del 0 al 255) dentro de él. Todas las operaciones matemáticas definidas en este ámbito resultarán un número de 8 bits.

La suma y la resta representan operaciones XOR y no hay diferencia entre adición y sustracción. Las multiplicaciones ya no son tan simples: Teniendo dos números de 8 bits a y b, y un producto p, también de 8 bits, los pasos que han de seguir serían:

Iterando con i=1 hasta 8:

Si el bit menos significativo de b es 1 entonces:

$p = p \oplus a$

Se realiza una copia de resguardo de a, luego:

$a = a << 1$

Si la copia de resguardo de a, en su bit más significativo, poseía un 1:

$a = a \oplus 0x1b$ (constante expresada en hexadecimal)

$b = b >> 1$

Terminado el ciclo de ocho iteraciones, p contendrá el valor resultado del producto entre a y b.

$\oplus$ Representa XOR
<< Desplazamiento a izquierda en la cantidad que especifique el operador de la derecha.
>> Desplazamiento a derecha en la cantidad que especifique el operador de la derecha.

Se comentó acerca de este proceso de multiplicación porque su inversa se utiliza en la generación de la S-box, Sustitution Box o caja de sustitución del algoritmo. A grandes rasgos, el múltiplo inverso de un número de entrada se almacenará en dos variables temporales, sobre las cuales se realizan operaciones de rotación, desplazamientos y XOR, para la obtención finalmente, del valor transformado.

2.4.9.4.IDEA

Internnational Data Enctyption Algorithm, o algoritmo Internacional de Encriptación de Datos, fue diseñado por Xuejia Lai y James L. Massey en el año 1991. Fue un algoritmo propuesto como reemplazo a DES.

Este algoritmo trabaja con bloques de 64 bits de longitud, con una llave de 128 bits. EL proceso para cifrar y descifrar es muy similar.

El concepto principal en cuanto al diseño del algoritmo radica en que se trata de una mezcla de operaciones de diferentes grupos algebraicos. Tres grupos forman parte de esta mezcla: EXOR, edición módulo 2^{16} y multiplicación en módulo 2^{16+1}.

En relación con cuestiones específicas del funcionamiento del algoritmo, tendremos en cuenta que trabaja con un bloque de entrada de 64 bits de longitud. Este bloque será dividido luego en cuatro sub-bloques de 16 bits cada uno: X1, X2, X3 y X4. Esto corresponderá a la entrada de la primera ronde del algoritmo. El algoritmo consta de ocho rondas en total. En cada una de estas rondas se realizan operaciones XOOR, adicionales y multiplicaciones sobre los cuatro sub-bloques, entre sí y con las seis sub-llaves de 16 bits. Entre rondas, el segundo y tercer sub-bloque son intercambiados. Por último, los cuatro sub-bloques son combinados con cuatro sub-llaves en una transformación de salida.

2.4.9.5.Blowfish

Ha sido desarrollado por Bruce Schneier en el año 1993. Fue ideado con la intención de reemplazar a DES, proponiéndolo como una alternativa sin problemas de patentamiento y libre de uso o de dominio libre. Se trata también, por supuesto, de un algoritmo criptográfico de llave simétrica y cifrado por bloques. Según opinión de expertos, para implementaciones actuales, quizá la mejor opción del algoritmo AES sea la conveniente.

El algoritmo consta de dos partes: La expansión de la llave y la encriptación de la información de entrada (que se procesará en bloques de 64 bits de longitud). A primera de estas partes se refiere a la conversión de la llave (de hasta 448 bits de longitud) en varios arrays o arreglos de sub-llaves. La segunda parte, respecto de la encriptación consiste en una función que se aplica 16 veces. Cada una de estas 16 iteraciones, o rondas, consistirá de una permutación dependiente de la llave y de una de sustitución dependiente, tanto de la llave como de la información de entrada. Casi todas las operaciones son adiciones y operaciones XOR en variables de 32 bits de longitud.

2.4.9.6.Twofish

Fue creado también por Bruce Schneier como *Blowfish* y si bien está relacionado con este último, es significativamente más complejo. El algoritmo vio la luz en el año 1998 y se trata también de un cifrador por bloques (ahora de 128 bits de longitud) que acepta una llave de longitud variable de hasta 256 bits. El algoritmo fue finalista entre los algoritmos propuestos para el AES, el concurso en que se estableció ganador al algoritmo *Rijndael*.

Las características distintivas del algoritmo podrían resumirse en que las S-boxes utilizadas serán dependientes de la llave y de un manejo de esta última relativamente complejo (una parte de la llave afectará al algoritmo de encriptación).

El algoritmo consta de una red Feistel que, realiza 16 rondas, aplicando una función biyectiva basada en cuatro S-boxes o cajas de sustitución de pendientes de la llave.

2.4.9.7.RC4

Es un algoritmo cifrador en flujo o stream cipher. *DES, TDES, IDEA, AES* y *Blowfish* son también algoritmos de criptografía simétrica, pero operan sobre bloques de entrada. Un cifrador en flujo operaría – esto no es así estrictamente en todos los casos, piénsese así de momento para

abordar el concepto- sobre cada byte de entrada... *RC4* es el algoritmo más popularmente utilizado entre los algoritmos de este tipo.

Fue desarrollado en el año 1984, por Ron Rivest para la compañía RC4 Data Security Inc. Que mantuvo en secreto el algoritmo hasta que en 1994 un anónimo lo hizo público en una lista de correo. Implementar RS4 de manera no oficial no es legal, pero no lo sería utilizar el nombre RC4. Es por esta razón que también se encontrará este algoritmo con nombres como ARC4 (ARCFOUR) o Alleged-RC4.

Trabaja en modo OFB[9].

El flujo de llave es independiente de la entrada o texto-plano. Consta de una S-box de 8x8, que referenciaremos más adelantes con los símbolos S0, S1….S255. Los elementos corresponden a una permutación de los números 0 a 255 y es una función -la permutación- de la llave de longitud variable. Contempla dos contadores que llamaremos i y j, inicializados en cero.

Para la generación de un byte aleatorio, los pasos son:

i = (i+1) mod 256
i = (i+Si) mod 256
Se intercambian Si y Sj
t = (Si+Sj) mod 256
K= St

El byte que contiene J y sobre el texto-plano se le aplica la operación XOR para producir el texto-cifrado; lo mismo si, en cambio se hiciera sobre el texto cifrado para producir el texto-plano.

La inicialización de la S-box se detalla de la siguiente manera. En primera instancia, se completará linealmente: S0=0, S1=1,…S255=255. Luego, se completará con otro array o arreglo de 256 bytes con la llave, repitiéndola tantas veces como sea necesario para llenarlo completamente: K0, K1,…, K255. Se establece el índice o contador j nuevamente a cero, para luego:

Iterando con i=0 hasta 255:

j = (j + Si + Ki) mod 256

[9] OFB: *Output feedback* o retroalimentación de salida. Este es un modo para utilizar un cifrador por bloques como un cifrador por flujo. La diferencia con respecto a CFB estriba en que aquí una cantidad determinada en bits de bloque de salida anterior se traslada hacia las posiciones menos significativas de la pila o la cola. Debe ser único pero no debe ser secreto.

Se intercambian Si y Sj

De esta manera finaliza el algoritmo. Una característica importante para tener en cuenta de este algoritmo es que es rápido; el proceso de encriptado es, en un orden aproximadamente de 10 veces, más rápido que el algoritmo DES.

2.4.9.8.Con funciones hash

2.4.9.8.1.MD5

El algoritmo de hashing criptográfico MD5 es una versión mejorada de su antecesor MD4. Las siglas MD corresponden a Message Digest o resumen de mensaje. Fueron diseñados por Ron Rivest (la "R" en RSA), quien luego de los resultados del cripto-análisis sobre MD4 por otros criptógrafos, parcialmente críticos, lo extendió; a esta versión extendida y más compleja la llamó MD5 y fue publicado en 1991.

El algoritmo produce, a partir de una entrada no limitada en cuanto a su tamaño, un hash criptográfico, o resumen de mensaje, de 128 bits de longitud. Se lo emplea actualmente en diferentes aplicaciones de seguridad, una de las más populares es la comprobación de integridad de archivos. Desde muchos sitios web que ofrecen descargas del archivo, se ofrece también el resultado hashing MD5 obtenido a partir del archivo, a manera de checksums o suma de comprobación, para poder confirmar, luego de descargarlo, que el hash criptográfico corresponde o no al publicado en el sitio Web. Estas comprobaciones suelen presentarse en cadena de 32 caracteres alfanuméricos, que corresponderán a los 16 bytes en formato hexadecimal.

En el año 1996, se descubrió un problema en el diseño de MD5. No se trataba de un problema, que volviese inseguras las implementaciones actuales, pero distintos expertos y criptógrafos propusieron la utilización de otros algoritmos de hashing criptográficos. Durante el año 2004, otros problemas más importantes fueron descubiertos y la seguridad del protocolo fue puesta

en tela de juicio. Por último, un grupo de investigadores describió, en el año 2007, como es posible la confección de dos archivos distintos que generan el mismo hashing MD5.

Ninguno de estos problemas citados implica que las implementaciones actuales de checksums o de registro de contraseñas mediante hashes (las dos aplicaciones más populares del algoritmo actualmente) se hayan vuelto vulnerables. Al menos por ahora, no se ha descubierto una técnica que permita obtener la entrada original a partir de hash o alterar a discreción un archivo y mantener un checksum o hash MD5 resultante. La recomendación, de cualquier manera, es la de la implementación de otro algoritmo como SHA-1.

Luego del preprocesamiento inicial, la entrada se procesa en bloques de 512 bits de longitud, divididos en 16 sub-bloques de 32 bits. La salida del algoritmo será una serie de cuatro bloques de 32 bits. Que concatenados representarán el hash criptográfico de 128 bits.

El mensaje o información de entrada es “paddeado”, o completado utilizando un padding, agregando un bit de valor 1 seguido de tantos bits en 0 como sea necesarios, de manera tal que la longitud resultante sea de 64 bits menor de un múltiplo de 512. Luego, se concatenará una presentación del tamaño o longitud original de la entrada utilizando 64 bits. Esto implica que, en esta instancia, el mensaje tendrá una longitud en bits múltiplo de 512.

El paso siguiente es la inicialización de cuatro variables de 32 bits, llamadas chaining variables, o variables de encadenamientos con valores prefijados:

A = 0x01234567
B = x089abcdef
C = 0xfedcba98
D = 0x76543210

Por cada bloque de 512 bits del mensaje, se producirá una iteración del ciclo principal del algoritmo. Dentro de éste, las cuatro variables son copiadas en otras cuatro variables temporarias.

a = A
b = B
c = C
d = D

Dentro de esta iteración principal, entonces se realizarán cuatro rounds o rondas. Cada ronda implementa una serie de operaciones, y la aplicación de una función distinta, 16 veces. Cada una de estas funciones corresponde a una no lineal, que se parametrizará sobre tres de estas últimas cuatro variables(a, b, c y d); cada vez que se adicionarán a este resultado la cuarta variable –la que no fue parámetro-, junto con un sub-bloque de entrada y una constante. Sobre este último resultado se realizará una rotación o desplazamiento hacia la derecha en una cantidad variable de bits y se sumará el resultado. Finalmente, el resultado reemplazará el contenido de alguna de estas cuatro variables.

Cuatro son las funciones no lineales preferidas; sólo una de ellas será utilizada en cada una de las cuatro rondas, en 16 ocasiones como hemos visto, donde en cada ocasión se combinarán de manera diferente los tres parámetros o variables de a, b, c y d. El símbolo $\oplus$ corresponde a la operación XOR, $\wedge$ a AND, $\vee$ a OR y $\neg$ a NOT.

$F(X, Y, Z) = (X \wedge Y) \vee ((\neg x) \wedge Z)$

$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$

$H(X, Y, Z) = X \oplus Y \oplus Z$

$I(X, Y, Z) = Y \oplus (X \wedge (\neg Z))$

Por último, luego de las cuatro rondas, de 16 pasos cada una a, b, c y d son sumadas A, B, C y D, y el algoritmo continuará con el siguiente bloque de entrada. Terminando el proceso del último toque, el resultado final corresponderá a la concatenación de A, B, C y D.

2.4.9.8.2.SHA

El NIST[10], en conjunto con la Agencia Nacional de Seguridad o National Security Agency (NSA) diseñaron el algoritmo SHA; éste fue nombrado a partir de las siglas del inglés Secure Hash Algorithm o Algoritmo de hash seguro.

[10] NIST: National Institute of Standards and Technology o Instituto Nacional de Tecnologías y Normalización

Fue diseñado por estos organismos para ser utilizado en otro estándar, el llamado DSS por Digital Signature Standard, que implementa el algoritmo DSA de firma digital.

Existen, en realidad, cinco versiones o variantes del algoritmo: SHA-1, que generará un hashing criptográfico de 160 bits, y las variantes que le siguieron, que se llaman en conjunto SHA-2 y corresponde al conjunto de algoritmos SHA-224, SHA-256, SHA-384 y SHA-512. Dentro de este último conjunto, el número que sigue a las siglas SHA especifica la longitud en bits del hashing criptográfico que genera el algoritmo.

El algoritmo, en mayor medida SHA-1, se implementa actualmente en diversas aplicaciones criptográficas. Como con MD5, quizá las más populares sean las del registro de hashes de contraseñas y la generación de checksums o comprobaciones de integridad de archivos.

Con respecto a la especificación del algoritmo, el sistema de padding es muy similar al del MD5, descripto en el apartado anterior. Luego, en lugar de cuatro variables de 32 bits con valores prefijados, SHA utiliza cinco –recuérdese que el algoritmo produce un hash criptográfico de 160 bits de longitud-, que se inicializan de la siguiente manera:

A = 0x67452301
B = 0xefcdab89
C = 0x98badcfe
D = 0x10325476
E = 0xc3d2e1f0

También, como en MD5, el proceso principal consta de un ciclo dentro del cual, en cada iteración, se procesarán 512 bits del mensaje o información de entrada. Una diferencia para notar es que cada iteración aplica una función no lineal, como MD5, pero 20 veces en lugar de 16. Las adiciones y el corrimiento o desplazamiento son similares a los de MD5.

Veamos cuáles son las funciones no lineales utilizadas en SHA, que aplicarán ahora sobre cinco variables temporarias (en lugar de cuatro) que serán los parámetros a, b, c, d y e. Como con lo anterior el símbolo

$\oplus$ Corresponde a XOR, $\wedge$ a AND, $\vee$ a OR y $\neg$ a NOT.

$f_t(X, Y, Z) = (X \wedge Y) \vee ((\neg x) \wedge Z)$ para t= 0 to 29

$f_t(X, Y, Z) = X \oplus Y \oplus Z$ para t= 20 to 39

ft (X, Y, Z) = (X ∧ Y) ∨ (X ∧ Z) ∨ (Y ∧ Z) para t= 40 to 59

ft (X, Y, Z) = X ⊕ Y ⊕ Z para t= 60 to 79

El numero t representa el número de operación, que estará entre 0 y 79. Recuérdese que se trata de cuatro rondas, de 20 operaciones que involucran a una de estas funciones, en cada uno de los pasos u operaciones.

De manera similar a MD5, pero con cinco variables en lugar de cuatro, el resultado final estará compuesto por la concatenación de A, B, C, D, E.

2.4.9.8.3.MAC (Códigos de autentificación de Mensajes)

Hablaremos, brevemente, sobre esta muy utilizada aplicación de las funciones de una vía o hashing. Los códigos de autenticación de mensajes, o MAC, por sus siglas en inglés Message Authentification Codes, implican el uso de funciones de una vía de hashing criptográfico vistas, pero incluyen una llave. Únicamente es utilizado para probar autenticidad de una información –un mensaje o un archivo- entre quienes conozcan la llave secreta.

Es posible aplicar esta técnica para autenticar archivos, frente a otros usuarios o, para un mismo usuario, para comprobar que el archivo haya sido alterado, de manera que puede ser utilizado también para pruebas de integridad (además de autenticidad entre usuarios, como hemos visto).

Una forma sencilla de convertir una función de hashing en un MAC es encriptar el resultado, el hash obtenido, con un algoritmo de encriptación simétrica. Sin embargo son más comunes las implementaciones que combinan la llave secreta con el mensaje, involucrando una función de hashing criptográfico.

Un algoritmo que implementa este protocolo es el HMAC.

2.4.9.8.4.HMAC

El algoritmo fue presentando en 1996 por Mihir Bellare, Ran Canetti y Hugo Krawczyk. Fue estandarizado en EE.UU. bajo el FIPS (Federal Information Processing Standard) como PUB 198.

HMAC puede ser utilizado con cualquier función de hashing, siendo las más comúnmente utilizadas MD5 y SHA-1. Cuando se utilice la primera, se especifica HMAC-MD5; en cambio, HMAC-SHA-1 implica que se usa SHA-1. HMAC-MD5 forman parte de la especificación de protocolos IPsec y TLS.

La definición formal del algoritmo es:

HMCk (m) = h ((K ⊕ opad) || h ((K ⊕ ipad) || m)

Donde h es la función de hashing criptográfico, K es llave secreta, “paddeada” o completada hacia la derecha con ceros al tamaño de bloque de la función de hashing (512 bits cuando se utiliza MD5 o SHA-1), m el mensaje para ser comprobado, || corresponde a la operación de concatenación, ⊕ a la operación XOR, el padding externo opad repite el byte 0x5c, y el interno ipad lo propio con el valor 0x36, por el tamaño de un bloque.

2.4.9.8.5.RSA

Es el algoritmo de cifrado asimétrico más popular en la actualidad. Creado por Ron Rivest, Adi Shamir y Leonard Adleman –nótese que son las iniciales de los apellidos que forman el nombre del algoritmo- fue publicado en el año 1977.Desde entonces ha resistido un extensivo cripto-análisis a través de años de los años, pero en realidad, no se ha comprobado matemáticamente que sea seguro; tampoco ha sido comprobado lo contrario, pero esto mismo sugiere a un nivel de confianza del algoritmo importante.

El algoritmo es considerado seguro, en tanto sean utilizadas llaves de longitud suficientemente seguras (se siguen utilizando llaves de 1024 bits, pero ya se recomienda al menos una longitud de 2048) e implementaciones actuales del algoritmo, donde se utilicen esquemas de padding seguros.

El algoritmo sirve para tanto para encriptar y desencriptar como para la generación de firmas digitales. Es, en la actualidad ampliamente utilizado en protocolos de comercio electrónico.

No hay que olvidar que debe usarse un padding criptográfico, tal como el definido en el estándar PKCS#1, para completar o "padder" el mensaje previamente a la encriptación. Así mismo, para la firma digital también debe tenerse en cuenta la utilización de un padding seguro, como el del esquema RSA-PSS, por ejemplo, y la misma llave no deberá utilizarse para ambas operaciones (la de encriptación y la de firma digital).

La seguridad está basada en la dificultad de realizar el factoreo de números grandes. La llave privada y la pública son generadas o calculadas en función de un par de números primos, del orden de los 200 dígitos o más grandes aún (en el RSA Factoring Challenge, el número RSA-200, por ejemplo correspondía a un número decimal de 200 dígitos, o 663 bits de llave, y fue factorizado en dos números decimales de 100 dígitos en el año 2005).

Al describir ya concretamente el algoritmo, se establece que para la generación del par de llaves (llave pública y llave privada) se deberán seleccionar dos números primos grandes aleatorios, p y q, y se calculará n como su producto:

$$n=pq$$

La llave de encriptación, e, será elegida también de manera aleatoria, tal que e y (p-1) (q-1) sean primos relativos[11].

La llave de desencriptación d será obtenida despejando la ecuación:

$$ed= 1 \bmod (p-1) (q-1)$$

En otras palabras, o mejor dicho, en otros símbolos:

[11] Nota: Dos números naturales se llaman primos relativos si el máximo común divisor entre ellos es 1.

$$d = e^{-1} \bmod ((p-1)(q-1))$$

Los números e y n componen la llave privada; el número d corresponde a la llave privada; p y q serán descartados pero no revelados.

A la hora de encriptar un mensaje m, éste deberá ser dividido en bloques más pequeños que n y cada parte del texto-cifrado, c, será obtenida mediante:

$$C_i = m_i^e \bmod n$$

Para la desencriptación, cada parte o bloque del texto-cifrado se tomará para calcular:

$$m_i = c_i^d \bmod n$$

2.4.9.8.6.ElGamal

Puede ser utilizado para encriptación y desencriptación de información, como para firma digital de documentos electrónicos.

El algoritmo fue descripto por Taher Elgamal en el año 1984. Está basado en otro algoritmo, el del acuerdo de llaves de Diffie-Hellman.

Este esquema está implementado actualmente en aplicaciones criptográficas muy populares como el software libre GNU Privacy Guard o GPG, y en versiones recientes de PGP.

A diferencia del algoritmo RSA, este algoritmo obtiene su seguridad a partir de la dificultad de calcular logaritmos discretos en un campo finito.

Resumiremos al algoritmo describiendo de manera rápida sus componentes principales, comenzando por la generación de las llaves públicas y privadas. Esto implicará la selección de un número primo p y dos números aleatorios, g y x, tales que g y x sean menores a p:

$$y=g^x \bmod p$$

La llave pública estará compuesta por y, g y p; mientras que x corresponderá a la llave privada.

Para la firma de un mensaje M, se elegirá un número aleatorio k, tal que éste sea un primo relativo a p-1 para calcular luego:

$$a=g^x \bmod p$$

Lo que sigue será la obtención de b a partir de la ecuación:

$$M= (xa+kb) \bmod (p-1)$$

La firma será entonces a y b y el valor de k se mantendrá secreto. Por último, la verificación de una firma deberá confirmar que:

$$y^a a^b \bmod p = g^M \bmod p$$

La utilización de ElGamal para la encriptación es muy similar al algoritmo de intercambio de llaves de Diffie-Hellman, salvo por un par de ligeras diferencias. Para encriptar un mensaje M, se deberá obtener un valor k de la misma manera que para la firma, pero luego se calculará:

a=gk mod p

Y

b= yk M mod p

Entonces a y b representarán al algoritmo cifrado.

2.4.9.8.7.DSA

Digital Signature Algorithm o algoritmo de Firma Digital, es una estándar del gobierno federal de los E.E.U.U., dentro de un Federal Information Processing Standard (FIPS).

Fu propuesto por el NIST, en el año 1991. La patente del algoritmo fue atribuida a David Kravitz, exempleado de la Agencia Nacional de Seguridad de los E.E.U.U. (NSA). El estándar fue llamado DSS (Digital Signature Standard).

El algoritmo de llave pública para la firma digital DSA es una variante de los algoritmos ElGamal y Schnorr.

La especificación del algoritmo se resumirá a continuación:

- p corresponderá a un número primo, de una longitud de 512 o 1024 bits (actualmente, estándar ha sido modificado, recomendando la utilización de 2048 bits), en tanto que esta longitud sea múltiplo de 64.

- q corresponderá a un número de 160 bits de longitud, factor primo de p-1.

- g será igual a $h^{(p-1)/q}$ mod p, donde h será cualquier número menor a p-1 tal que $h^{(p-1)/q}$ mod p sea mayor que 1.

- x corresponderá a un número menor que q.

- y será igual a g^x mod p.

De esta forma, el número x representará la llave privada y la llave pública será y.

Para la firma de un mensaje m, se deberá obtener un número aleatorio k, tal que éste sea menor a q, y calcular.

r = (gk mod p) mod q
Y

s = (k-1(H (m) +xr)) mod q

Nótese que H (m) es una función de hashing; el estándar especifica la implementación del algoritmo de hashing criptográfico SHA-1- los números calculados r y s reasentarán la firma.

La verificación de la misma se realizaría calculando.

w = s-1 mod q
u1 = (H(m)*w) mod q
u2 = (rw) mod q
v = ((gu1 * gu2) mod p) mod q

Entonces, si v resultase igual a r, la firma será verificada.

2.4.9.8.8.Diffie Hellman

Si bien el algoritmo no puede ser utilizado para la encriptación y desencriptación de información, permitirá a dos partes, sin conocimiento previo la una de la otra, establecer conjuntamente una llave secreta compartida sobre un canal o medio inseguro. Típicamente, esta llave secreta será luego utilizada para encriptar las comunicaciones sobre el canal inseguro, mediante el uso de criptografía simétrica. Se debe recordar que el algoritmo se utiliza para el intercambio o distribución de llaves: Dos partes podrán utilizarlo para la generación de una llave secreta compartida, pero no para encriptar o desencriptar mensajes.

Fue el primer algoritmo de llave pública conocido, diseñado por Whitfield Diffie y Martin Hellman en el año 1976. La seguridad del algoritmo está basada en la dificultad de calcular logaritmos discretos en un campo finito, en comparación con la facilidad de calcular la exponenciación en el mismo campo finito.

En cuanto a la especificación matemática del algoritmo, las dos partes acordarán un número primo grande n, y un número g, tal que este último sea primitivo base mod n. Estos dos enteros no son secretos, lo que quiere decir que las dos partes pueden acordarlos sobre el canal inseguro. El

protocolo continúa de la siguiente manera: La parte “A” elige un número aleatorio grande, x, y envía a la parte “B” lo que sigue:

$$X= g^x \bmod n$$

La parte “B” también elegirá un número aleatorio grande, y, y enviará a la parte “A”:

$$Y= g^y \bmod n$$

La parte “A” calculará:

$$k=y^x \bmod n$$

Lo propio hará la parte “B”, calculando:

$$k´= X^y \bmod n$$

Entonces tanto k como k´ serán iguales a $g^{xy} \bmod n$. Ninguna persona –o computadora- que haya monitoreado el canal inseguro puede calcular ese valor, ya que solo conocería n, g, X e Y. dada la dificultad del cálculo del logaritmo discreto (n debe ser un número grande, en esto radica en parte la seguridad del sistema) para recuperar x o y, la llave secreta generada, k, puede ser compartida por las partes de manera segura.

3. DESARROLLO DE LA TESINA

3.1. Descripción del Sistema

Se empleará como ejemplo práctico para esta Tesina, un sistema reducido el cual será una aplicación web con la que se obtendrá un formulario de ingreso, uno de registro (los usuarios solo pueden ser creados por usuarios del sistema), una interfaz principal y una más con la posibilidad de hacer consulta/alta/baja y modificación, la cual demostrará la metodología propuesta en esta Tesina. La aplicación estará montada sobre un servidor web Tomcat, el cual deberá tener un servidor de correo con el cual se enviará un mail con la llave para ingresar al sistema y crear la sesión la cual autenticará a la persona en el sistema para poder trabajar en él.

3.2. Lenguajes de Programación

A continuación se realizará una breve descripción de tres de los lenguajes de programación más utilizados a nivel web y se justificará el porqué de la elección de uno de ellos.

3.2.1.Elección del Lenguaje de Programación

3.2.1.1.Java[12]:

"Un lenguaje simple. Orientado al objeto, distribuido, interpretado, sólido, seguro, de arquitectura neutral, portable, de alto desempeño, de multihilos y dinámico"

- **Simple**

Basado en el lenguaje C++ pero donde se eliminan muchas de las características OOP (Programación orientada a objetos) que se utilizan esporádicamente y que creaban frecuentes problemas a los programadores. Esta eliminación de causas de error y problemas de mantenimiento facilita y reduce el coste del desarrollo de software.

- Java no da soporte a struct, union y pointer.
- Java no ofrece typedef ni #define.
- No permite la sobrecarga de operadores.
- No ofrece herencia múltiple.
- Maneja los comandos en línea de diferente manera que C++.
- Java tienen una clase String, que permite un mejor manejo que los arrays de terminación nula del C y C++.

[12] Datos extraídos de: http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html

- Java tiene un sistema automático de asignación y liberación de memoria (recolector de basura) que mejora mucho los sistemas del C++.

• **Orientado al objeto:** Java da buen soporte a las técnicas de desarrollo OOP y en resumen a la reutilización de componentes de software.

• **Distribuido:** Java se ha diseñado para trabajar en ambiente de redes y contienen una gran biblioteca de clases para la utilización del protocolo TCP/IP, incluyendo HTTP y FTP. El código Java se puede manipular a través de recursos URL con la misma facilidad que C y C++ utilizan recursos locales (archivos).

• **Interpretado:** El compilador Java traduce cada fichero fuente de clases a código de bytes (Bytecode), que puede ser interpretado por todas las máquinas que den soporte a un visualizador de que funcione con Java. Este Bytecode no es específico de una máquina determinada, por lo que no se compila y enlaza como en el ciclo clásico, sino que se interpreta.

• **Sólido:** El código Java no se quiebra fácilmente ante errores de programación. Así el relaje que existe en la declaración y manejo de tipos en C y C++ se torna en restricciones en Java, donde no es posible la conversión forzada (cast) de enteros en punteros y no ofrece soporte a los punteros que permitan saltarse reglas de manejo de tipos. Así en Java no es posible escribir en áreas arbitrarias de memoria ni realizar operaciones que corrompan el código. En resumen se eliminan muchas de las posibilidades de "trucos" que ofrecía el C y C++.

• **Seguro:** Como Java suele funcionar en ambiente de redes el tema de seguridad debe interesar en sobremanera. Las mismas características antes descritas que evitan la corrupción de código evitan su manipulación. Actualmente se está trabajando en encriptar el código.

• **Arquitectura neutral:** El compilador crea códigos de byte (Bytecode) que se envía al visualizador solicitado y se interpreta en la máquina que posee un intérprete de Java o dispone de un visualizador que funciona con Java.

• **Portable:** Al ser de arquitectura neutral es altamente portable, pero esta característica puede verse de otra manera: Los tipos estándares (int, float, etc.) están igualmente implementados en todas las máquinas por lo que las operaciones aritméticas funcionaran igual en todas las máquinas.

• **Multihilos:** Java puede aplicarse a la realización de aplicaciones en las que ocurra más de una cosa a la vez. Java, apoyándose en un sistema de gestión de eventos basado en el

paradigma de condición y monitores C.A.R.[13] permite apoyar la conducta en tiempo real e interactiva en programas

En la programación concurrente, un monitor es una construcción que permite la sincronización de subprocesos tienen tanto la exclusión mutua y la capacidad de esperar (bloque) para una determinada condición sea verdadera. Los monitores también disponen de un mecanismo para la señalización de otros hilos que indica que su condición se ha cumplido. En esencia, un monitor M = (m, c) es un par de un mutex (bloquear) objeto m y una condición variable de c. Una variable de condición es básicamente un contenedor de hilos que están a la espera de una determinada condición. Monitores proporcionan un mecanismo para subprocesos para abandonar temporalmente el acceso exclusivo a fin de esperar a que alguna condición que deben cumplir, antes de recuperar el acceso exclusivo y reanudar su tarea.

- Otra definición de monitor es una clase, objeto, o módulo de subprocesos que utiliza la exclusión mutua envuelta con el fin de permitir el acceso a un método o variable por más de un hilo de forma segura. La característica definitoria de un monitor es que sus métodos son ejecutados con exclusión mutua: En cada punto en el tiempo, a lo sumo un thread puede estar ejecutando cualquiera de sus métodos. Los Monitores fueron inventados por CAR Hoare y por Brinch Hansen, y se llevaron a cabo por primera vez en el lenguaje Pascal concurrente de Brinch Hansen.

- **Dinámico:** al contrario que C++ que exige se compile de nuevo la aplicación al cambiar una clase madre Java utiliza un sistema de interfaces que permite aligerar esta dependencia. Como resultado, los programas Java pueden permitir nuevos métodos y variables en un objeto de biblioteca sin afectar a los objetos dependientes.

3.2.1.2.ASP.NET

ASP.NET[14] es un marco de trabajo de programación generado en Common Language Runtime[15] que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

[13] Concepto y fragmento sobre C.A.R extraído de http://en.wikipedia.org/wiki/Monitor_(synchronization)
[14] Fuente: http://mysf.galeon.com/segunda.htm#¿Qué es ASP.NET

• Mejor rendimiento. ASP.NET es un código de Common Language Runtime compilado que se ejecuta en el servidor. A diferencia de sus predecesores, ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.

• **Compatibilidad con herramientas de primer nivel:** El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de Visual Studio. La edición WYSIWYG, los controles de servidor de arrastrar y colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta.

• **Eficacia y flexibilidad**: Se basa en Common Language Runtime, la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de Acceso a datos se encuentran accesibles desde el Web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes. Además, la interoperabilidad de Common Language Runtime garantiza que la inversión existente en programación basada en COM se conserva al migrar a ASP.NET.

• **Simplicidad:** Facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios. Por ejemplo, el marco de trabajo de página de ASP.NET permite generar interfaces de usuario, que separan claramente la lógica de aplicación del código de presentación, y controlar eventos en un sencillo modelo de procesamiento de formularios de tipo Visual Basic.

• **Facilidad de uso:** Emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web. Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Esta filosofía de "administración local cero" se extiende asimismo a la implementación de las aplicaciones ASP.NET Framework. Una aplicación ASP.NET Framework se implementa en un servidor sencillamente mediante la copia de los archivos necesarios al servidor. No se requiere el

---

[15] .NET Framework proporciona un entorno en tiempo de ejecución denominado Common Language Runtime, que ejecuta el código y proporciona servicios que facilitan el proceso de desarrollo. Los compiladores y las herramientas exponen su funcionalidad y permiten escribir código con las ventajas que proporciona este entorno de ejecución administrado. Fuente: http://msdn.microsoft.com/es-es/library/8bs2ecf4(v=vs.110).aspx

reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.

- **Escalabilidad y disponibilidad:** ASP.NET se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

- **Posibilidad de personalización y extensibilidad:** ASP.NET presenta una arquitectura bien diseñada que permite a los programadores insertar su código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado. La implementación de la autenticación personalizada o de los servicios de estado nunca ha sido más fácil.

- **Seguridad**: Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.

Para que la aplicación Web atienda las solicitudes, ASP.NET debe analizar y compilar primero el código de la aplicación Web en uno o varios ensamblados. Cuando se compila el código, se traduce en una representación independiente del lenguaje y de la CPU llamado Lenguaje intermedio de Microsoft (MSIL). En tiempo de ejecución, MSIL se ejecuta en el contexto de .NET Framework, que traduce MSIL en instrucciones específicas de la CPU para el procesador en el equipo que ejecuta la aplicación.

La compilación dinámica de ASP.NET permite modificar el código fuente sin tener que compilarlo explícitamente antes de implementar la aplicación web. Si modifica un archivo de código fuente, ASP.NET lo vuelve a compilar automáticamente y actualiza todos los recursos vinculados. No es necesario reiniciar el servidor IIS (Internet Information Server) para que los cambios se apliquen, a no ser que se modifique la sección <processModel>[16].

Puede extender el sistema de compilación de ASP.NET creando proveedores de compilación personalizados para nuevos tipos de archivo que se invoquen durante la compilación.

- **Compilar durante la primera solicitud:** De forma predeterminada, las páginas Web ASP.NET y los archivos de código se compilan de forma dinámica la primera vez que los

[16] Configura las opciones del modelo de procesamiento de ASP.NET en un servidor Web con los Servicios de Microsoft Internet Information Server (IIS). La sección **processModel** sólo se puede establecer en el archivo Machine.config, y afecta a todas las aplicaciones ASP.NET que se ejecutan en el servidor. Fuente: http://msdn.microsoft.com/es-es/library/7w2sway1(v=vs.85).aspx

usuarios solicitan un recurso, como una página de ASP.NET (archivo. aspx), de un sitio Web. Una vez compiladas las páginas y los archivos de código por primera vez, los recursos compilados se almacenan en la caché para que las siguientes veces que solicite la misma página sea lo más eficaz posible.

- **Compilación dinámica de páginas ASP.NET**: (archivos .aspx), servicios Web ASP.NET (archivos .asmx), controladores HTTP de ASP.NET (archivos .ashx) y archivos de aplicación ASP.NET (Global.asax), además de otros archivos, como código fuente y archivos de clases.

- **Volver a compilar al efectuar cambios:** Cualquiera cambio efectuado en un archivo compilado dinámicamente invalidará de forma automática el ensamblado compilado almacenado en caché del archivo y activará el volver a compilar de todos los recursos afectados. La próxima vez que se realice una solicitud en el código, ASP.NET reconocerá que el código ha cambiado y volverá a compilar los recursos afectados de la aplicación Web. Este sistema le permite desarrollar rápidamente las aplicaciones con una carga de procesamiento de compilación mínima. (Tenga en cuenta que las implicaciones de los cambios realizados en los recursos pueden oscilar entre tener que volver a compilar una sola página y volver a compilar todo el sitio web).

Al compilar el código, los ensamblados resultantes se almacenan en memoria caché en una carpeta del servidor. Esta carpeta requiere los permisos adecuados para que el código se compile y se ejecute correctamente. Puede configurar la ubicación de la carpeta de compilación y los permisos con los que se compila y funciona el código.

De forma predeterminada, al compilar una aplicación Web, el código compilado se coloca en la carpeta de archivos temporales de ASP.NET. Esta carpeta es un subdirectorio de la ubicación en la que instaló .NET Framework. Normalmente, la ubicación es la siguiente:

%SystemRoot%\Microsoft.NET\Framework\versionNumber\Temporary ASP.NET

3.2.1.3.PHP

Hypertext Preprocessor, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. PHP inicio como una modificación a Perl escrita por Rasmus Lerdorf a finales de 1994. Su primer uso fue el de mantener un control sobre quien visitaba su curriculum en su web.

Traduciendo la definición del FAQ de PHP.net:

"PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas".

Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI[17], como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Un sitio con páginas dinámicas es el que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para requisitos articulares. Las aplicaciones dinámicas para el Web son frecuentes en los sitios comerciales (e-commerce), donde el contenido visualizado se genera de la información alcanzada en una base de datos u otra fuente externa.

Soporte para bases de datos:

Una de sus características más potentes es su suporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse InterBase, mSQL, MySQL, Oracle, Informix, PosgreSQL, entre otras. PHP también ofrece la integración con las varias bibliotecas externas, que permiten que

---

[17] **Interfaz de entrada común** (en inglés *Common Gateway Interface*, abreviado **CGI**) es una importante tecnología de la World Wide Web que permite a un cliente solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de **CGIs**.
Las aplicaciones CGI fueron una de las primeras prácticas de crear contenido dinámico para las páginas web. En una aplicación CGI, el servidor web pasa las solicitudes del cliente a un programa externo. Este programa puede estar escrito en cualquier lenguaje que soporte el servidor, aunque por razones de portabilidad se suelen usar lenguajes de script. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional.
CGI ha hecho posible la implementación de funciones nuevas y variadas en las páginas web, de tal manera que esta interfaz rápidamente se volvió un estándar, siendo implementada en todo tipo de servidores web.

el desarrollador haga casi cualquier cosa desde generar documentos en PDF hasta analizar código XML.

Su sintaxis es muy similar a la del ASP, pues el código PHP va incrustado dentro del código HTML. Sus tags van incluidos dentro de <?php ?>. Un ejemplo práctico de una instrucción funcional de PHP sería:

<?php

print "Hola, mundo!";

?>

Que al ser ejecutado en el servidor nos imprimiría dentro del código HTML la frase:

Hola, mundo!

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores.

Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. Es utilizado en aplicaciones Web-relacionadas por algunas de las organizaciones más prominentes tales como Mitsubishi, Redhat, Der Spiegel, MP3-Lycos, Ericsson y NASA.

PHP es la opción natural para los programadores en máquinas con Linux que ejecutan servidores web con Apache, pero funciona igualmente bien en cualquier otra plataforma de UNIX o de Windows, con el software de Netscape o del web server de Microsoft. PHP también utiliza las sesiones de HTTP, conectividad de Java, expresiones regulares, LDAP, SNMP, IMAP, protocolos de COM (bajo Windows).

Para trabajar con capacidades PHP, se puede conseguir mayor información en PHP.net, sitio encargado de mantener al día a todos los desarrolladores con las últimas descargas relacionadas con el lenguaje y documentación.

3.2.2.Ventajas y Desventajas

En esta sección lo que se busca es realizar un listado breve de lo que son algunas ventajas y desventajas de los lenguajes previamente citados.

Java:

Ventajas

- **Portabilidad**: Un solo código funciona para todos los browsers compatibles con Java o donde se tenga una Máquina Virtual de Java (Mac's, PC's, Sun's, etc.). Debido a la JVM, un browser compatible con Java deberá ejecutar cualquier programa hecho en Java, esto ahorra a los usuarios tener que estar insertando "plug-ins" y demás programas que a veces nos quitan tiempo y espacio en disco.

- Las páginas de Web, ya no tienen que ser estáticas, se le pueden poner toda clase de elementos multimedia y permiten un alto nivel de interactividad, sin tener que gastar en paquetes carísimos de multimedia.

- El JDK es una herramienta libre de licencias (sin costo), creada por Sun.- Está respaldado por un gran número de proveedores.

- Debido a que existen diferentes productos de Java, hay más de un proveedor de servicios.

- Sun saca al mercado cada 6 meses una nueva versión del JDK.

- Es independiente de la plataforma de desarrollo.

- Existen dentro de su librería clases gráficas como awt y swing, las cuales permiten crear objetos gráficos comunes altamente configurables y con una arquitectura independiente de la plataforma.

• Se puede acceder a bases de datos fácilmente con JDBC[18], independientemente de la plataforma utilizada. El manejo de las bases de datos es uniforme, es decir transparente y simple.

• Existen las herramientas Crystal Reports o herramientas libres como iText que los genera en formato PDF. La API que utilizan estas herramientas en Java, es la más recomendable para generar reportes en Web.

• **Simple**. Elimina la complejidad de los lenguajes como "C" y da paso al contexto de los lenguajes modernos orientados a objetos.

• **Robusto**. El sistema de Java maneja la memoria de la computadora por ti. No te tienes que preocupar por apuntadores, memoria que no se esté utilizando, etc. Java realiza todo esto sin necesidad de que uno se lo indique.

• **Seguro**. El sistema de Java tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no puede hacer con los recursos críticos de una computadora.

• **Portable**. Como el código compilado de Java (conocido como byte code) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java.

• **Independiente a la arquitectura**. Al compilar un programa en Java, el código resultante un tipo de código binario conocido como byte code. Este código es interpretado por diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida.

• **Multithreaded**. Un lenguaje que soporta múltiples threads es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.

• **Interpretado**. Java corre en máquina virtual, por lo tanto es interpretado.

[18] Java Database Connectivity (JDBC) API es el estándar industrial para la conectividad de base de datos independiente entre el lenguaje de programación Java y una amplia gama de bases de datos, Bases de datos SQL y otras fuentes de datos tabulares, como hojas de cálculo o archivos planos. El API JDBC proporciona una API de nivel de llamada de acceso de base de datos basada en SQL. Fuente: http://www.oracle.com/technetwork/java/javase/jdbc/index.html

• **Dinámico**. Java no requiere que compiles todas las clases de un programa para que este funcione. Si realizas una modificación a una clase Java se encarga de realizar un Dynamic Binding o un Dynamic Loading para encontrar las clases.

Java puede funcionar como una aplicación sola o como un "applet", que es un pequeño programa hecho en Java. Los applets de Java se pueden "pegar" a una página de Web (HTML), y con esto puedes tener un programa que cualquier persona que tenga un browser compatible podrá usar.

Desventajas

• **Velocidad**: Los programas hechos en Java no tienden a ser muy rápidos, Como los programas de Java son interpretados nunca alcanzan la velocidad de un verdadero ejecutable.

• Java es un lenguaje de programación. Esta es otra gran limitante, por más que digan que es orientado a objetos y que es muy fácil de aprender sigue siendo un lenguaje y por lo tanto aprenderlo no es cosa fácil. Especialmente para los no programadores. En conclusión su curva de aprendizaje es muy larga.

• Pero en general Java posee muchas ventajas y se pueden hacer cosas de escalas muy grades.

• Hay diferentes tipos de soporte técnico para la misma herramienta, por lo que el análisis de la mejor opción se dificulta.

• Para manejo a bajo nivel deben usarse métodos nativos, lo que limita la portabilidad.

• El diseño de interfaces gráficas con awt y swing no es simple. Existen herramientas como el JBuilder que permiten generar interfaces gráficas de manera sencilla, pero tienen un costo adicional.

• Puede ser que no haya JDBC para bases de datos poco comerciales.

• Algunas herramientas tienen un costo adicional

PHP[19]:

Ventajas:

- Multiplataforma
- Manejo de excepciones
- Biblioteca nativa de funciones
- Permite técnicas de programación orientada a objetos.20
- Amplia documentación en su página oficial-> PHP
- Destacada conectividad con MySQL.
-
- Es libre.

Desventajas:

- Promueve creación de código desordenado y con un mantenimiento complejo.
- No posee adecuado manejo de Unicode.
- Es muy difícil de optimizar.
- Diseñado especialmente hacia un modo de realizar aplicaciones Web que es problemático y obsoleto.

ASP.NET[21]

Ventajas

- Se encarga de detectar el tipo de navegador utilizado por el cliente a la hora de realizar una petición al servidor y en consecuencia determinar la versión HTML que éste soporta.

[19] Datos extraídos de: http://blogs.utpl.edu.ec/disenowebymultimedia/2009/07/23/ventajas-y-desventajas-de-php-2/
[20] A partir de PHP 5. Referencia: http://www.php.net/manual/es/oop5.intro.php
[21] Fuente: http://www.linkedin.com/company/1078501/advantages-disadvantages-of-making-website-in-asp-net-537455/product

- Es liviano.

- Se puede utilizar en cualquier computadora que esté conectada a la red que tenga instalado un servidor IIS.

- Línea de aprendizaje breve.

- Tiene la facilidad de conectarse con la base de datos, que hace que sea más fácil.

- Velocidad de uso. Se pueden desarrollar sitios de manera muy rápida e intuitivamente.

- Controles Avanzados: ASP.Net viene con una serie de colecciones de elementos ricos de controles servidor y cliente que se pueden. Lo bueno de estos controles es que la mayoría de ellos se puede utilizar de inmediato utilizar para desarrollar grillas, gestores, calendarios, etc. Lo bueno de estos controles es que la mayoría de ellos se puede utilizar de inmediato.

- Seguridad: viene incluido con un cierto grado de seguridad que permite soportar autorizaciones, autentificaciones y otras opciones de implementación como Kerberos, NTLM o algún otro estándar. Además Con la configuración por aplicación y la autentificación de Windows es posible obtener una seguridad más completa de las aplicaciones.

Desventajas

- **Unidades de testeo no son performantes:** ASP.NET tiende a no soportar herramientas de testeo.

- **Vistas de estado afectan el rendimiento:** las vistas de estado pueden algunas veces tomar efectos negativos en el rendimiento o performance, especialmente con controles complejos del servidor. Esto también puede ser muy grande durar mucho tiempo.

- **Demasiadas Ventanas.** Algunos desarrolladores sienten que ASP.NET intenta seguir la forma de Windows haciendo demasiado de cerca algo torpe para las aplicaciones basadas en web.

- **Costos:** Los sitios desarrollados en ASP necesitan correr en servidores con IIS el cual aumenta el costo de contratación y mantenimiento del mismo. ASP[22] No es pago lo que sí es pago es el entorno de desarrollo y el IIS. Pero también existe: "Visual Web Developer Express"

[22] Fuente: http://social.msdn.microsoft.com/Forums/es-ES/e50e3d1d-6dc5-4521-a973-8f4d41544000/aspnet-es-gratis-o-tiene-algn-costo?forum=netfxwebes

que es gratuita.

3.2.3.Lenguaje Elegido

Como conclusión de los puntos anteriores en donde se evaluó las ventajas, desventajas y característica de algunos lenguajes de programación se opta por JAVA como lenguaje de programación basado en experiencias previas en proyectos personales, es un lenguaje que se estudió en la universidad, la amplia comunidad que aporta código, mejoras, reportes de problemas y soluciones en toda la web, el mismo está muy difundido e instaurado en el mercado y posee toda la arquitectura y robustez necesaria para poder realizar desde sistemas chicos hasta desarrollos empresariales, sumado a la gran escalabilidad que posee.

3.3. Base de Datos

Todo sistema puede o no dejar persistente cierta información que es de carácter primordial para el mismo, siempre se debe seleccionar correctamente “que” almacenar y “que no”. Hoy en día se utilizan las bases de datos como medio de persistencia de la información.

Una base de datos[23] o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

[23] http://es.wikipedia.org/wiki/Base_de_datos

3.3.1.Comparativa

Algunos fragmentos de la siguiente comparativa es tomada y traducida del paper: “Journal of Computer Science & Research (JCSCR) - ISSN 2227-328X http://www.jcscr.com Vol. 1, No. 1, Pages. 20-31, February 2012”.

3.3.2.Reseñas

En esta sección se realiza una breve reseña de algunos de los diferentes SGBD evaluados para utilizar en la Tesina, ellos son MS SQL Server 2008, Oracle 11g, MySQL 5.5 e Informix, seguido de un cuadro comparativo y posterior elección del mismo.

3.3.3.MS SQL Server 2008

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) producido por Microsoft. Su lenguaje de consulta primaria es Transact-SQL, una aplicación de la norma de consulta estructurado ANSI / ISO Language (SQL) utilizado por Microsoft y Sybase. Microsoft SQL Server es compatible con transacciones atómicas, consistentes, aisladas y duraderas. Incluye soporte para la creación de reflejo de base de datos y la agrupación. Un servidor SQL clúster es una colección de servidores configurados de forma idéntica, que ayudan a distribuir la carga de trabajo entre múltiples servidores. SQL Server también es compatible con la partición de datos para bases de datos distribuidas. Además de la creación de una imagen de base de datos que permite la creación de espejos del contenido de bases de datos, junto con los registros de transacciones, en otra instancia de SQL Server, está basado en ciertos factores desencadenantes predefinidos.

3.3.4.Oracle 11g

Oracle[24] 11g ofrece un rendimiento y una escalabilidad excepcionales en servidores Windows, Linux y UNIX, y aporta un rápido rendimiento de la inversión porque permite pasar de un solo servidor a Grid Computing sin modificar ni una sola línea de código. Oracle 11g automatiza las tareas de administración y ofrece las mejores funciones de seguridad y de cumplimiento de las normativas, por lo que consigue resultados óptimos. Gracias a Real Aplicación Clusters, se obtienen los mayores niveles de disponibilidad. Como ofrece distintas ediciones y unos costes operativos más bajos que IBM DB2 y Microsoft SQL Server, es la opción ideal para empresas en expansión. Compare las distintas ediciones para saber cuál es la que más le conviene.

Comprende al menos una instancia de la aplicación, junto con el almacenamiento de datos. Una instancia comprende un conjunto de procesos del sistema operativo y memoria estructuras que interactúan con el almacenamiento. Además del almacenamiento, la base de datos se compone de registros de logs en línea que mantienen el historial de transacciones. Los procesos pueden a su vez archivar los registros de logs en los registros de almacenado, que proporcionan la base para la recuperación de datos y para algunas formas de replicación de datos. Los almacenes de datos Oracle RDBMS lógicamente en la forma de la tabla de espacios y físicamente en forma de archivos de datos. A nivel físico, los archivos de datos comprenden uno o más bloques de datos, en los que el tamaño de bloque puede variar entre los archivos de datos. Oracle cuenta con el diccionario de datos, índices, y racimos.

3.3.5.MySQL 5.5

MySQL ofrece una, multi-usuario, multi-hilo, muy rápido y robusto de SQL (Structured Query Language) servidor de base de datos. Servidor MySQL está diseñado para sistemas de

[24] Párrafo extraído de http://www.oracle.com/es/solutions/midsize/oracle-products/database/index.html

producción de misión crítica, alta carga de trabajo así como para integrarse en software para ser distribuido.

MySQL tiene una doble licencia. Los usuarios pueden optar por utilizar el software MySQL como un producto Open Source bajo los términos de la Licencia Pública General de GNU (http://www.fsf.org/licenses/) o pueden adquirir una licencia comercial estándar de Oracle. Ver http://www.mysql.com/company/legal/licensing/

3.3.6.Informix[25]

IBM Informix es un software de base de datos relacional, integrable y de fácil uso. Es uno de los servidores de bases de datos más utilizadas del mundo, con los usuarios que van desde las corporaciones más grandes del mundo a pymes. Informix incorpora conceptos de diseño que son significativamente diferentes de plataformas relacionales tradicionales, niveles extremadamente altos de rendimiento y disponibilidad, capacidades distintivas en la replicación de datos y escalabilidad, y gastos administrativos mínimos.

3.3.7.Comparación[26]

Tabla de comparación de los diferentes DBMS

DMBS CARACTERÍSTICAS

[25] *http://www-01.ibm.com/software/data/informix/*

[26] Comparación obtenida de: https://es.scribd.com/doc/109205352/Tabla-de-comparacion-de-los-diferentes-DBMS

Oracle 11g	- Almacena datos en tablas relacionales - Ayuda a mejorar la eficiencia y el rendimiento de los datos almacenados - Entorno cliente/servidor. - Gestión de grandes bases de datos, Usuarios concurrentes y Alto rendimiento en transacciones. - Sistemas de alta disponibilidad, Disponibilidad controlada de los datos de las aplicaciones.- Gestión de la seguridad. - Autogestión de la integridad de los datos. - Portabilidad, Compatibilidad y Conectividad. - Rendimiento alto y escalabilidad (permite a los usuarios acceder rápida y eficientemente a la información). - La seguridad se cubre con las medidas como: la autentificación, la autorización, y la encriptación. - Costos elevados y en dólares.
SQL Server 2008	- Confianza de misión crítica con mayor tiempo activo. - Rendimiento ultra rápido y características mejoradas de seguridad para cargas de trabajo de misión crítica. - Capacidades interactivas de visualización de datos. - Programabilidad, Manejabilidad y Alta disponibilidad básica. - Inteligencia de negocios de auto-servicio (Alertas). - Integración de datos avanzada (Agrupamiento y búsqueda. difusa, Captura de cambios a datos, Data Mining avanzado). - Seguridad avanzada (SQL Server Audit, Cifrado transparente de datos)- Almacenamiento de datos (Índice ColumnStore, Compresión, Particiones). - Alta disponibilidad avanzada (Múltiples secundarios activos; Geo-Clustering, Multi-sitios,). - Integración con Internet (incluye compatibilidad integrada con XML)- Escalabilidad y disponibilidad - Facilidad de instalación, distribución y utilización. - Incluye herramientas para extraer y analizar datos de resumen para el procesamiento analítico en línea. - Costos medios y en dólares.
Informix	- Confiable, Bajo costo –en dólares-, No presenta complicaciones, Mejor

	integración al mercado. - Capacidad de relación de datos en múltiples lugares físicos - Ocupa menos recursos y memoria que Oracle, Mayor costo en relación a SQL Server - Se integra con Linux, Oracle y otras BD - Conecta datos relacionales con páginas web - Facilita generación de aplicaciones orientadas a internet - Seguridad y restauración de alta velocidad - Lento en cuanto a rendimiento frente a los otros DBMS - Basado en UNIX, siendo más sólido
MySQL 5.5	- Soporta gran cantidad de tipos de datos para las columnas. - Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos. - Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla. - Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo. - Flexible sistema de contraseñas y gestión de usuarios, con un muy buen nivel de seguridad en los datos. - Open Source, es gratis siempre y cuando el código sea open source, sino es un producto de costo medio y en dólares.

3.3.8.Elección del motor de base de datos

Por motivos de licenciamiento –open source-, experiencia trabajando con él desde la facultad hasta desarrollos encarados en forma personal y particular, que posee la robustez necesaria para realizar desde un trabajo de investigación hasta proyectos empresariales los cuales pueden llegar a manejar Terabytes de datos se elige MySQL como motor de base de datos.

3.3.9.Otros de talles técnicos a considerar

Algunas consideraciones de las más importantes que se deben tener antes de realizar algún sistema es el entorno donde va a correr el mismo, también si va a ser multiplataforma, multiusuario, multi-hilo, entre otras.

El sistema a desarrollar en la Tesina será multiusuario ya que varios usuarios podrán ingresar e utilizarlo en forma simultánea.

Será multiplataforma ya que JAVA es multiplataforma debido a su máquina virtual y como el sistema de ejemplo de la Tesina se desarrollará en él y en entorno web del mismo, también se lo considera multiplataforma porque se ejecuta en los navegadores convencionales como lo son Firefox, Chrome o Internet Explorer que todos son multiplataforma. No garantiza que en todos los navegadores funcione completamente ya que cada uno al actualizarse va variando conforme a sus desarrollos.

El ejemplo de desarrollo que se tomará en la Tesina correrá en entorno web, el sistema se montará sobre un servidor web[27] Apache[28] Tomcat versión 8. Utilizando Java Standard Edition versión 7 y el Java Development Kit (JDK) 1.7.

3.4. Desarrollo del sistema

El sistema a desarrollar a modo de demostración de ésta Tesina consiste en una aplicación web de envergadura chica con énfasis en la seguridad.

---

[27] Servidor web: Programa que implementa el protocolo HTTP de la capa de aplicación del modelo OSI – Generalmente sobre el puerto 80.

[28] Apache: es un servidor HTTP de código abierto y multiplataforma que surgió en 1995. Desde 1996 es el servidor HTTP más usado, alcanzó el 70% del mercado. Apache Tomcat, es un servidor web con soporte Java Servlets y Java Server Pages.
Datos tomados de “¿Cómo elegir un servidor web?” por Juan F. Belón Pérez de la Escuela Técnica Superior de Ingenierías

Se desarrollará un sistema que gestione usuarios con diferentes permisos y el enfoque estará en el código para la validación del usuario.

Básicamente consta de una interfaz de login –acceso al sistema ingresando usuario y contraseña-, una página principal, una de datos personales, otra para gestionar usuarios y para finalizar una de gestión o administración de perfiles, sumada a una última interfaz para ver los accesos que posee el tipo de usuario.

En el sistema existen 4 tipos diferentes de usuarios:

- ***Administrador***: Quien es el de mayor jerarquía en la aplicación puede hacer todo.

- ***Supervisor:*** Se encarga de controlar lo realizado por los usuarios estándares y son los que puede eliminar y editar lo cargado.

- ***Usuario Estándar***: Puede dar de alta y consultar el sistema.

- ***Consulta:*** Solo puede realizar consultas de los datos de los usuarios, no puede ni actualizar ni cargar.

Luego de realizar la validación de usuario y contraseña el sistema generará una clave (llave) generada por un algoritmo desarrollado para esta Tesina que se detalla a continuación:

El sistema realiza los cálculos para generar una llave única con la existencia de 2 grupos de información, una pertinente al usuario y otra a la fecha y hora actual:

Grupo 1:

- *Usuario.*
- *Nombre.*
- *Apellido.*
- *Mail.*
- *DNI.*

Grupo 2:

- *Hora.*
- *Minuto.*
- *Segundo.*
- *Fecha en formato (dd/mm/aaaa).*

A partir de esos datos el sistema el sistema genera la llave ITERANDO 4 veces, concatenando el resultado final del procedimiento de:

- Con la función random de 0 a 4 toma elemento de grupo 1. En el caso de ser 0 toma una cadena vacía “”;
    - Siendo:
        - 0 - Usuario.
        - 1 - Nombre.
        - 2 - Apellido.
        - 3 - Mail.
        - 4 - DNI.
- Con la función random de 0 a 3 toma elemento de grupo 2. En el caso de ser 0 toma una cadena vacía “”;
    - Siendo:
        - 0 - Hora.
        - 1 - Minuto.
        - 2 - Segundo.
        - 3 - Fecha en formato (dd/mm/aaaa).
- Concatena ambas.
- Iterar N veces (n=4)

Una vez finalizado este procedimiento encripta en SHA-1 y lo almacena en la base de datos como llave para luego ser enviada al usuario vía mail, quien tendrá 3 intentos de acceso con esta clave, si la equivoca 3 veces el usuario será BLOQUEADO. En caso afirmativo poseerá acceso al sistema.

3.4.1.Límites

El sistema evidenciará principalmente un acceso al sistema NO contemplará:

- Un log.
- Edición de interfaces.
- Edición de temas para el usuario.
- Edición de Nuevos menús.
- Creación de nuevos perfiles.
- Creación de nuevos permisos para los perfiles.

- No se tendrá acceso a los datos de las sesiones.

3.4.2.Alcances

El sistema contendrá como contenido mínimo ya que se utilizará de modelo para evidenciar una metodología:

- Alta, baja, modificación y consulta de usuarios.
- Usuarios eliminados
- Filtros por usuario, nombre, apellido y DNI.
- Edición y consulta de perfiles.
- Consulta de usuarios conectados.
- Consulta de Datos personales.
- Un administrador podrá desconectar a los usuarios.
- Cambio de contraseña.

3.4.3.Diagrama de Casos de Uso

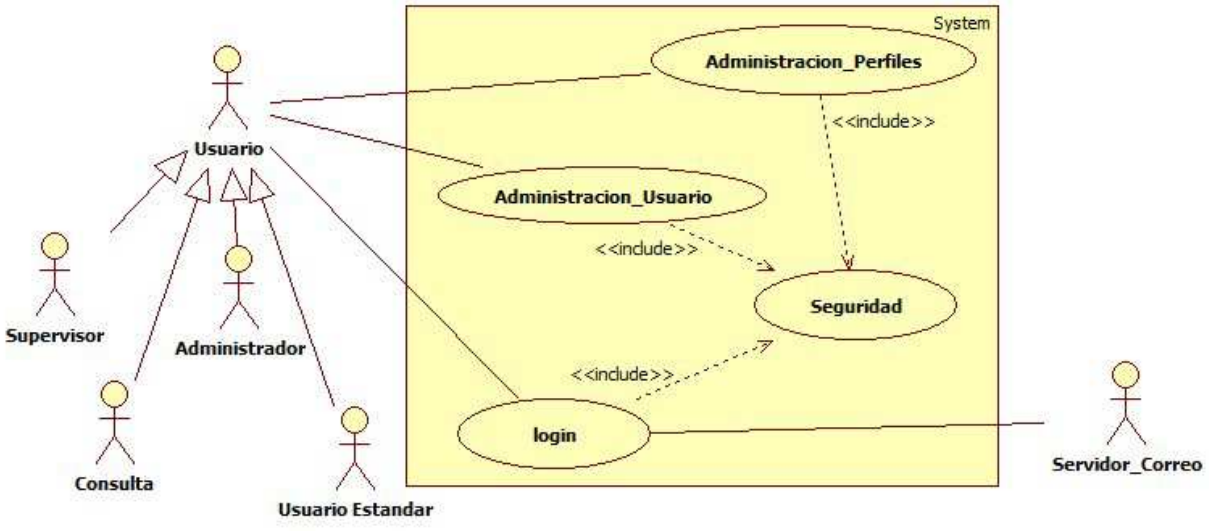


Ilustración 1 - Diagrama de Casos de uso

3.4.4.Especificaciones

3.4.4.1.Caso de Uso

La forma de documentación del sistema a desarrollar fueron tomados del curso de UML dictado por la empresa local SOLUS tomado en 2012.

Característica	Descripción
Código	CU001
Nombre	Administración de Usuarios
Paquete	Proyecto_Tesis::Documentación
Versión	1.0
Descripción	Altas, Bajas, Modificación y Consulta de usuarios.
Autor	Matías Sosa
Complejidad	Media
Prioridad	Alta
Importancia	Alta
Observaciones	Es de carácter importante la validación de Mail porque es el eje del sistema.

Característica	Descripción
Código	CU002
Nombre	Administración de Perfiles
Paquete	Proyecto_Tesis::Documentación
Versión	1.0
Descripción	Consulta de Perfiles y asignación de elementos de menú al perfil con sus respectivos permisos.
Autor	Matías Sosa
Complejidad	Media
Prioridad	Alta
Importancia	Alta
Observaciones	--

Característica	Descripción
Código	CU003
Nombre	Login
Paquete	Proyecto_Tesis::Documentación
Versión	1.0
Descripción	Usuario accede al sistema y debe pasar por él ya que es la parte de validación más importante y el núcleo de la tesina
Autor	Matías Sosa
Complejidad	Baja
Prioridad	Alta
Importancia	Crítica
Observaciones	

Característica	Descripción
Código	CU004
Nombre	Seguridad
Paquete	Proyecto_Tesis::Documentación
Versión	1.0
Descripción	Todas las pautas sobre seguridad del sistema.
Autor	Matías Sosa
Complejidad	Critica
Prioridad	Alta
Importancia	Alta
Observaciones	--

3.4.4.1.1.Actores

Nombre	Abst.	Cant.	Descripción breve
Usuario	Si	N	Es el que administra el sistema.
Administrador	No	N	Es el usuario que tiene permiso total sobre la aplicación. Es el único que puede limpiar contraseñas y puede ver que usuarios están conectados.
Supervisor	No	N	Es el usuario que se encarga de controlar la actividad de los usuarios estándares. No puede cargar el sistema, pero si modificar y eliminar.
Usuario_Estandar	No	N	Es el usuario que se encarga de cargar el sistema, no puede modificar ni eliminar.
Consulta	No	N	Es el usuario que puede ver lo que hay, tiene ciertas restricciones como no ver usuarios eliminados y demás. No puede cargar, ni editar ni eliminar.
Servidor_Correo	No	N	Son los servidores con los cuales se interactuará para poder enviar el mail de validación los más comunes son : Hotmail, Gmail y Yahoo!

3.4.4.1.2.Cursos

3.4.4.1.2.1. Normal

CUS003		Referencias
1	El usuario abre un navegador web ingresa al sitio.	
2	El sistema solicita los siguientes datos: • Usuario • Contraseña	
3	El usuario ingresas sus datos.	
4	El sistema valida que los datos ingresados sean incorrectos.	• {CA001}
5	El sistema realiza los cálculos para generar una llave única con los siguientes datos: Grupo 1: • Usuario. • Nombre. • Apellido. • Mail. • DNI. Grupo 2: • Hora. • Minuto. • Segundo. • Fecha en formato (dd/mm/aaaa). A partir de esos datos el sistema el sistema genera la llave iterando 4 veces, concatenando el resultado final del procedimiento: • Con la función random de 0 a 4 toma elemento de grupo 1. En el caso de ser 0 toma una cadena vacía “”; o Siendo: ▪ 0 - Usuario. ▪ 1 - Nombre. ▪ 2 - Apellido. ▪ 3 - Mail. ▪ 4 - DNI. • Con la función random de 0 a 3 toma elemento de grupo 2. En el caso de ser 0 toma una cadena vacía “”; o Siendo: ▪ 0 - Hora. ▪ 1 - Minuto. ▪ 2 - Segundo. ▪ 3 - Fecha en formato (dd/mm/aaaa).	

	• Concatena ambas. • Iterar N veces (n=4 en este caso) Una vez finalizado este procedimiento encripta en SHA-1 y lo almacena en la base de datos como llave.	
6	El sistema envía la llave generada en el punto 5 al mail registrado en la base de datos del usuario que desea loguearse. El mail contiene {RN015}>	
7	El sistema muestra una ventana emergente (popup) solicitando que ingrese la llave.	
8	El usuario busca en su mail -con el cual se lo registró al sistema-, copia la llave y lo pega en la ventana	
9	El sistema valida la clave.	• {CA002}
10	El sistema almacena los siguientes datos de la sesión en la tabla de sesión: • hora_ingreso – hora exacta en la que el usuario ingresó la llave de verificación – en formato hh-mm-ss • llave_sesion - la llave generada y enviada al usuario • estado_id – El id de estado, que es validado • observación – Algún comentario extra para agregar, generalmente esta en valor “”. • fecha - fecha del día en formato dd/mm/aaaa • tiempo_sesion – El tiempo máximo que puede durar la misma. • Identificador univoco del usuario	
11	El Caso de Uso termina cuando el sistema presenta la pantalla de inicio del sistema, mostrando los menús de: • Inicio {SF001} • Gestión de Usuarios o Datos Personales {SF002} o ABM Usuarios {SF003} • Seguridad o Perfiles {SF004} o Ver Usuarios Conectados {SF005} • Cerrar Sesión {SF006} En caso de que el TIPO de USUARIO.{RN012} no posea los permisos necesarios se enviará a la página de permiso denegado o insuficientes.	• {SF001} • {SF002} • {SF003} • {SF004} • {SF005} • {SF006} • {RN012}

3.4.4.1.2.2. Subflujos

Sf001	**Inicio / Página principal**	
N°	**Paso**	**Referencias**
1	El caso de uso termina cuando el sistema muestra una página que contiene una bienvenida al sistema.	

Sf002	**Datos Personales**	
N°	**Paso**	**Referencias**
1	El Sistema muestra una pantalla donde se pueden apreciar todos los datos que posee el usuario.	

2	El usuario puede revisar sus datos desplegando las ventanas y ocultando las que no desee ver.	
3	Si el usuario desea cambiar la contraseña {SF007}	• {SF007}
4	El subflujo finaliza cuando el usuario cierra sesión o cambia de página.	

Sf003	**ABM usuarios**	
N°	**Paso**	**Referencias**
1	El sistema presenta una lista de todos los usuarios del sistema según restricción de permisos.	
2	El subflujo termina cuando usuario elije: • Alta {SF008} • Baja {SF009} • Editar {SF010} • Ver {SF011} • Cambiar de página	• {SF008} • {SF009} • {SF010} • {SF011}

Sf004	**Perfiles**	
N°	**Paso**	**Referencias**
1	El sistema presenta la interfaz mostrando dos listas: • Lista de Perfiles: Con botón “ELEGIR” o Tipo Usuario (Seleccionable) • Lista de permiso que tiene cada perfil: o Menú | Permiso {RN013} | Estado {RN014} • Botones de Editar y de Ver	• {RN013} • {RN014}
2	El usuario selecciona un elemento de la lista con los permisos que posee cada perfil	
3	El subflujo termina cuando: • El usuario da click al botón editar {SF012} • El usuario da click al botón ver {SF011} • El usuario cambia de interfaz	

Sf005	**Ver Usuarios Conectados**	
N°	**Paso**	**Referencias**
1	El sistema presenta la interfaz mostrando una lista con los usuarios conectados al momento de ingresar en la página mostrando los datos: • Lista de Perfiles: Con botón “ELEGIR” o Usuario o Apellido o Nombre o ID Sesión o Fecha o Hora Ingreso	
2	EL usuario puede elegir: • Actualizar la lista • Desconectar usuario	
3	El caso de uso termina si el usuario elige actualizar actualiza la lista.	
4	El caso de uso termina Si el usuario elige desconectar usuario pregunta por sí o por no y se actualiza la lista en caso afirmativo.	

Sf006	Cerrar Sesión	
N°	**Paso**	**Referencias**
1	El usuario selecciona el botón cerrar sesión.	
2	El sistema muestra el cartel diciendo: • ¿Seguro Desea Cerrar Sesión?	
3	El usuario selecciona SI.	
4	El sistema almacena en la tabla de sesión la hora de egreso del sistema y todos los datos necesarios.	
5	El sistema redirige a la página de login.	

Sf007	Cambio de Contraseña	
N°	**Paso**	**Referencias**
1	El usuario da click en el botón cambiar contraseña	
2	El sistema despliega una interfaz que le permite ingresar: • Contraseña Anterior. • Contraseña Nueva. • Repetir Contraseña Nueva.	
3	El sistema valida los datos	• {CA003}
4	El sistema guarda la contraseña nueva.	
5	El subflujo termina cuando el sistema muestra el cartel de contraseña guardada.	

Sf008	Alta de Usuario	
N°	**Paso**	**Referencias**
1	El usuario da click en el botón alta de usuario	
2	El sistema valida que tenga permisos. {RN012}	• {RN012}
3	El sistema despliega una interfaz popup modal donde posee los datos que debe cargar: • Usuario • Contraseña • Confirma Contraseña • DNI • Nombre • Apellido • Mail • Mail Alternativo • Combo con los Estados existentes • Combo con los Tipos de Usuario existentes	
4	El usuario carga completa los datos y le da click a guardar.	
5	El sistema valida los datos.	• {RN001} • {RN002} • {RN003} • {RN004} • {RN005} • {RN006} • {RN007} • {RN008} • {RN009} • {RN010}
6	El Subflujo termina cuando el sistema almacena los datos del usuario nuevo y cierra la ventana modal.	

Sf009	Baja de Usuario	
N°	**Paso**	**Referencias**
1	El usuario da click al botón baja de usuario. {RN011}	• {RN011}
2	El sistema valida que tenga permisos. {RN012}	• {RN012}
3	El sistema muestra cartel preguntando: • ¿Está seguro que desea eliminar el usuario?	
4	El usuario selecciona si	
5	El Subflujo termina cuando el sistema guarda el estado del usuario como “Eliminado”	

Sf010	Editar de Usuario	
N°	**Paso**	**Referencias**
1	El usuario da click al botón Editar de usuario. {RN011}	• {RN011}
2	El sistema valida que tenga permisos. {RN012}	• {RN012}
3	El sistema muestra la interfaz con los datos del usuario: • Usuario • DNI • Nombre • Apellido • Mail • Mail Alternativo • Combo con el Estados que tiene asignado • Combo con el Tipo de Usuario que tiene asignado	
4	El usuario gestor edita los datos que necesite actualizar.	
5	El sistema valida los datos	• {RN001} • {RN004} • {RN005} • {RN006} • {RN007} • {RN008} • {RN009} • {RN010}
6	El usuario pulsa el botón guardar	
7	El subflujo termina cuando el sistema almacena los cambios y cierra la ventana.	

Sf011	Ver Usuario	
N°	**Paso**	**Referencias**
1	El usuario da click en ver usuario	
2	El sistema despliega una ventana con los datos – en forma de texto - : • Usuario • DNI • Nombre • Apellido • Mail • Mail Alternativo • Texto con la descripción del Estado que tiene asignado • Texto con la descripción del Tipo de Usuario que tiene asignado	
3	El usuario da click en la cruz (“X”) para cerrar la venta.	
4	El subflujo termina cuando El sistema cierra la ventana.	

Sf012	Editar Tipo Menú Permiso	
N°	Paso	Referencias
1	El usuario da click en el botón editar	
2	El sistema valida que tenga permisos. {RN012}	• {RN012}
3	El sistema despliega una ventana con los datos: • Tipo de Usuario: <tipo> • Estado: <Con valor seleccionado: Estado traído del elemento seleccionado> • Permiso: <Con valor seleccionado: Permiso traído del elemento seleccionado>	
4	El usuario realiza las modificaciones necesarias.	
5	El usuario da click en el botón Guardar.	
6	El subflujo termina cuando El sistema guarda los datos y cierra la ventana.	

3.4.4.1.2.3. Cursos Alternativos

CA001	Datos Invalidos	
N°	Paso	Referencias
1	El sistema no puede validar los datos	
2	El sistema muestra un cartel de usuario o contraseña incorrecta	
3	El caso de uso alternativo concluye cuando el sistema devuelve el foco al campo de usuario.	

CA002	Llave Inválida	
N°	Paso	Referencias
1	El sistema valida la llave y da un error.	
2	El sistema muestra un cartel de que no se pudo validar la llave y que la vuelva a ingresar.	
3	Si ingresa bien sigue curso, pero si ingresa mal la llave por tercera vez ir a {CA004}	• {CA004}
4	SI ingresa bien la llave el caso de uso termina ingresando al sistema y retornando al punto 10 del {CU003}.	

CA003	Contraseñas no coinciden	
N°	Paso	Referencias
1	El sistema valida la contraseña ingresada con el campo re ingresar contraseña.	
2	El sistema muestra un cartel de que las contraseñas no coinciden.	
3	El caso de uso alternativo termina cuando el sistema posiciona el foco en el campo contraseña.	

CA004	Se ingresó 3 veces mal la contraseña	
N°	Paso	Referencias
1	El sistema valida la llave por tercera vez y da un error.	
2	El sistema cambia el estado del usuario a BLOQUEADO.	
3	El caso de uso termina cuando el sistema cierra la ventana de la llave y muestra el cartel con el aviso del bloqueo.	

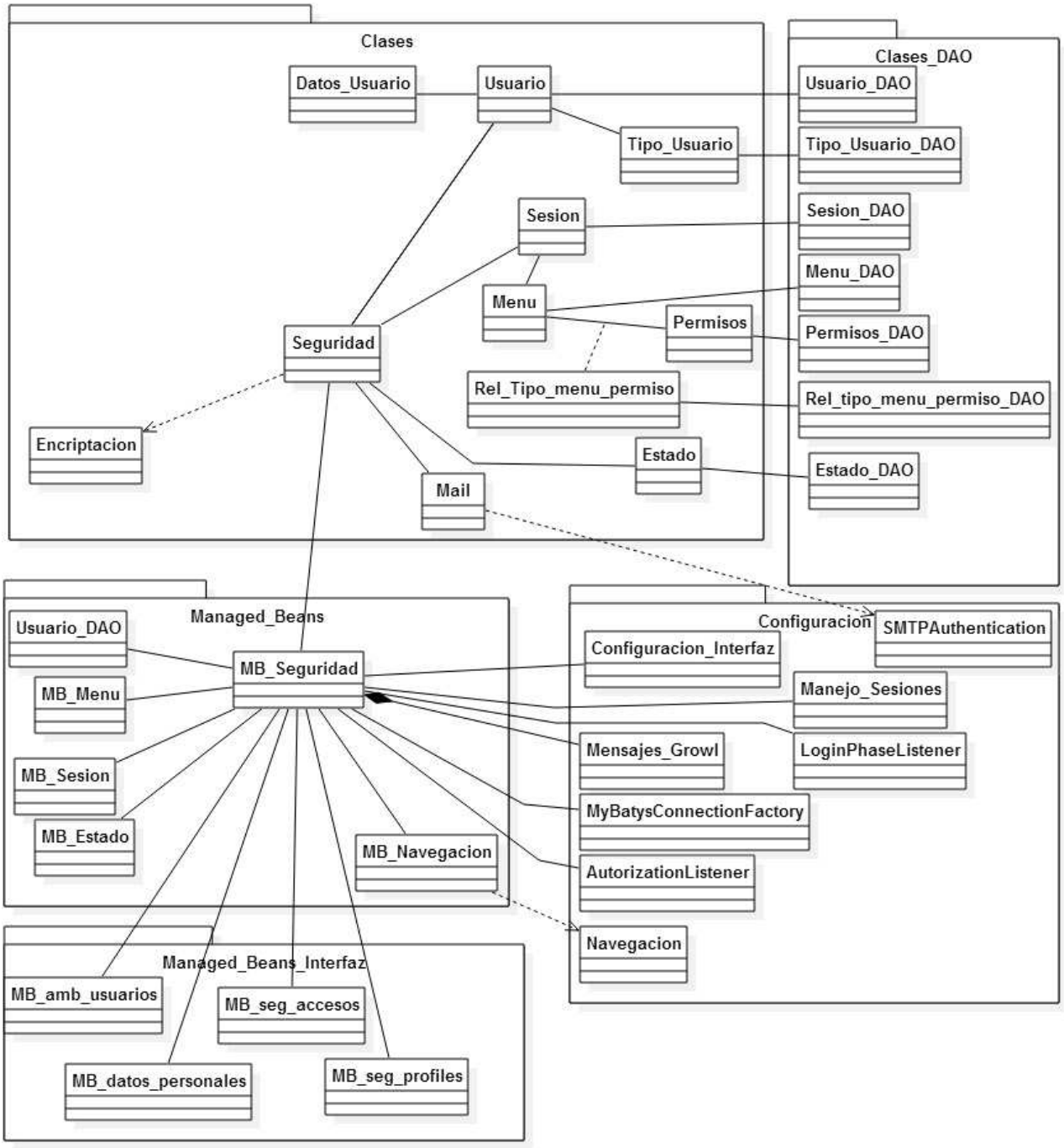
3.4.4.1.3.Reglas de Negocio

Regla	Nombre	Descripción	Valor actual
RN001	Valida_Usuario	El usuario es el identificador que posee la persona dueña del mismo, la cual lo utiliza como pseudónimo para ingresar al sistema	• NO puede ser un valor vacío “”. • No debe existir en el sistema 2 usuarios con la misma nomenclatura • Largo máximo de la cadena es de 255. • No se permiten caracteres especiales, salvo “_” y “-“.
RN002	Valida_Contraseña	La contraseña es una clave personal, que no debe difundirse, ni si quiera contarse, por cuestiones de seguridad que le permite a la persona ingresar junto con su usuario al sistema.	• NO puede ser un valor vacío “”. • Largo mínimo es de 8 caracteres. • Es Alfanumérico • Se encripta en SHA-1 al almacenado. • Se guarda en la base de datos en SHA-1de una vía.
RN003	Valida_Confirma _Contraseña	La validación de contraseña sirve para tratar de que la persona NO se equivoque al momento de ingresar una contraseña de acceso al sistema. Es imposible evitar errores, esto sirve para mitigarlos.	• NO puede ser un valor vacío “”. • Largo mínimo es de 8 caracteres. • Es Alfanumérico • Se encripta en SHA-1al almacenado. • NO se guarda en la base de datos. • Debe coincidir con la Contraseña.
RN004	Valida_DNI	Documento único de identidad. Sirve para identificar a una persona en la República Argentina.	• NO puede ser un valor vacío “”. • No se deben permitir valores alfabéticos • No se deben permitir NINGUN carácter especial • Cantidad máxima de caracteres es de 10 dígitos
RN005	Valida_Nombre	Nombre del propietario del usuario.	• NO puede ser un valor vacío “”. • Debe ser carácter alfabético. • Cantidad máxima de caracteres permitidos es de 255. • No puede contemplar caracteres especiales.
RN006	Valida_Apellido	Apellido del propietario del usuario	• NO puede ser un valor vacío “”. • Debe ser carácter alfabético. • Cantidad máxima de caracteres permitidos es de 255. • No puede contemplar caracteres especiales.
RN007	Valida_Mail	Es el correo con el cual interactuará el sistema para enviar la clave. No puede ser duplicado y en lo posible debe ser de los servidores de correo más convencionales.	• NO puede ser un valor vacío “”. • NO pueden existir 2 mail registrados en el sistema • Debe existir un validador de formato de mail: que contemple “@ y “.com”, “.com.ar”, “.es”, etc. • Cantidad máxima de caracteres es de 255. • No se permiten caracteres especiales, salvo “_”, “-“y “.”
RN008	Valida_Mail_lternati vo	Sirve para poseer una comunicación alternativa en caso de algún error o problema con el mail principal. NO se tomará como mail principal sino como	• SI puede ser un valor vacío “”. • Debe existir un validador de formato de mail: que contemple “@ y “.com”, “.com.ar”, “.es”,

		alternativo en casos problemáticos	etc. • Cantidad máxima de caracteres es de 255. • No se permiten caracteres especiales, salvo “_”, “-“y “.”
RN009	Valida_cmb_ Estados	El estado en el que puede estar el usuario. El usuario gestor de otros usuarios debe ver ID-Descripcion_Estado.	• NO puede ser un valor vacío “”. • No puede quedar en valor “Seleccione un…” • Debe ser elegido desde un campo combo
RN010	Valida_cmb_tipo	El Tipo de usuario que debe ser el usuario. El usuario gestor de otros usuarios debe ver ID-Descripcion_Tipo_Usuario.	• NO puede ser un valor vacío “”. • No puede quedar en valor “Seleccione un…” • Se debe seleccionar el valor desde un combo
RN011	Usr_Seleccionado	Debe haber seleccionado un elemento de la lista	• Debe haber una de las filas seleccionadas para poder trabajar.
RN012	Permisos_Tipo_Usuario	El sistema debe validar que tipo de usuario es y con eso ver que permisos tiene sobre el sistema	• Usuario: ADMINSITRADOR puede: o Dar de alta usuarios o Dar de baja Usuarios o Modificar Usuarios. o Consultar Usuarios o Ver usuarios eliminados o Modificar perfiles de otros usuarios que NO sean administradores. o Consultar accesos. • Usuario: Supervisor o Dar de baja Usuarios o Modificar Usuarios. o Consultar Usuarios o Ver usuarios eliminados o Modificar perfiles de otros usuarios que NO sean administradores. o Consultar accesos. • Usuario: Estándar o Dar de alta usuarios o Consultar Usuarios o Modificar perfiles de otros usuarios que NO sean administradores. o Consultar accesos. • Usuario: Consulta o Consultar Usuarios. o Consultar accesos.
RN013	Permisos	Son todos los permisos que van a guardarse en la base de datos y las distintas combinaciones que puede adoptar, para poder utilizarse en la tesis.	Identificador – Descripción del Permiso • 0 - Ningún Permiso • 1 - Consulta • 2 - Consulta + Modificación • 3 - Consulta + Baja • 4 - Consulta + Alta • 5 - Consulta + Alta + Baja • 6 - Consulta + Alta + Modificación • 7 - Todos los Permisos • 8 - Consulta + Baja + Modificación
RN014	Estados	Son los diferentes estados por los que pueden pasar algunos de los elementos que componen al sistema.	Identificador – Estado • 1 - Habilitado • 2 - Deshabilitado • 3 - Pendiente • 4 – Eliminado
NR015	Mail	Es el mail que se enviará al usuario con la clave.	• Asunto: Clave para Ingreso a Sistema

			Tesis • Mensaje: Para poder Ingresar al sistema debe copiar el siguiente código (sin las comillas): "<llave codificada generada en punto 4>" Una vez ingresada podrá acceder al sistema. Si se desloguea, pierde la sesión o cierra su navegador deberá loguearse nuevamente e ingresar la clave generada nuevamente.

3.4.5.Diagrama de Clases



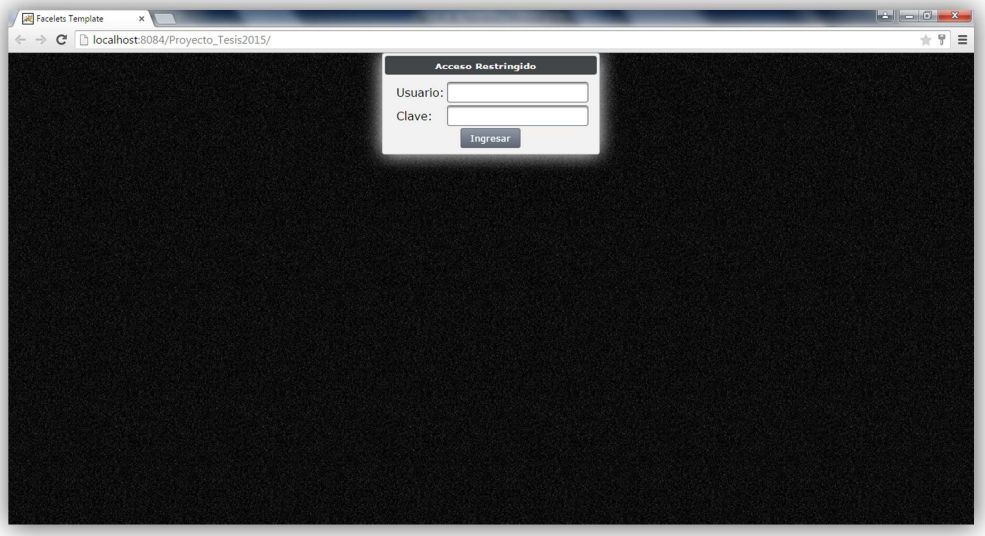
Es un diagrama de clases conceptual debido a restricciones de espacio, donde se obvian métodos y atributos los cuales están a disposición en el código del sistema.

3.4.6.Interfaces de Usuario de la Aplicación

3.4.6.1.1.Ingreso al Sistema

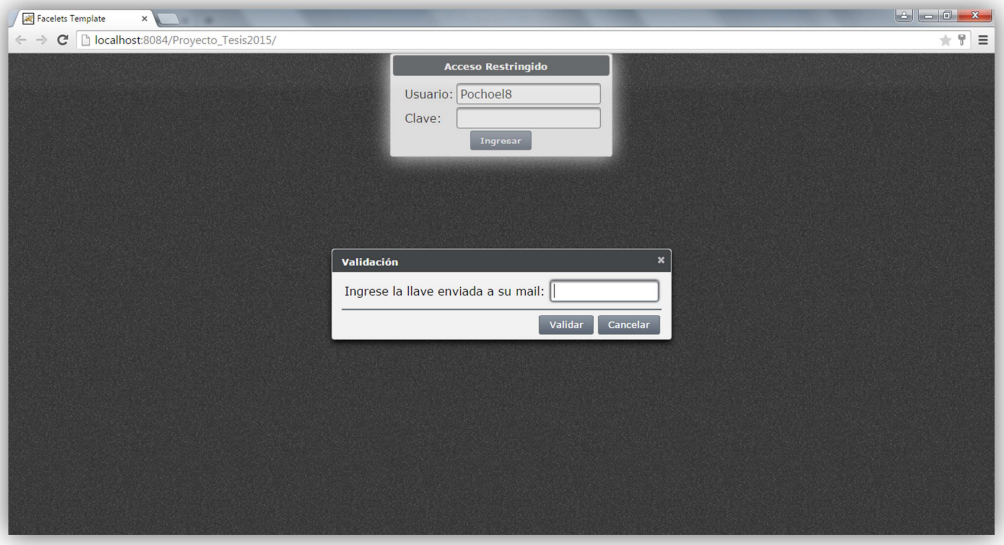
3.4.6.1.1.1. Ingreso Normal al Sistema

El acceso al sistema comienza con un login que solicita Usuario y Contraseña.



Captura de la pantalla de login del sistema

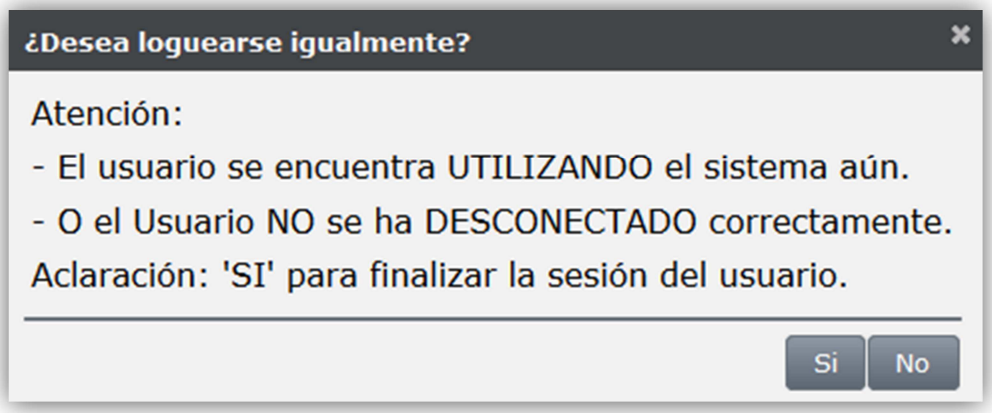
El sistema comienza con un login que solicita Usuario y Contraseña.



Captura de la pantalla que solicita el ingreso de la llave enviada al mail

3.4.6.1.1.2. Ingreso Alternativo al Sistema

El ingreso alterno se produce cuando el sistema detecta que ya existe un usuario conectado. Por ende el usuario puede estar conectado o simplemente NO haber cerrado la sesión correctamente.

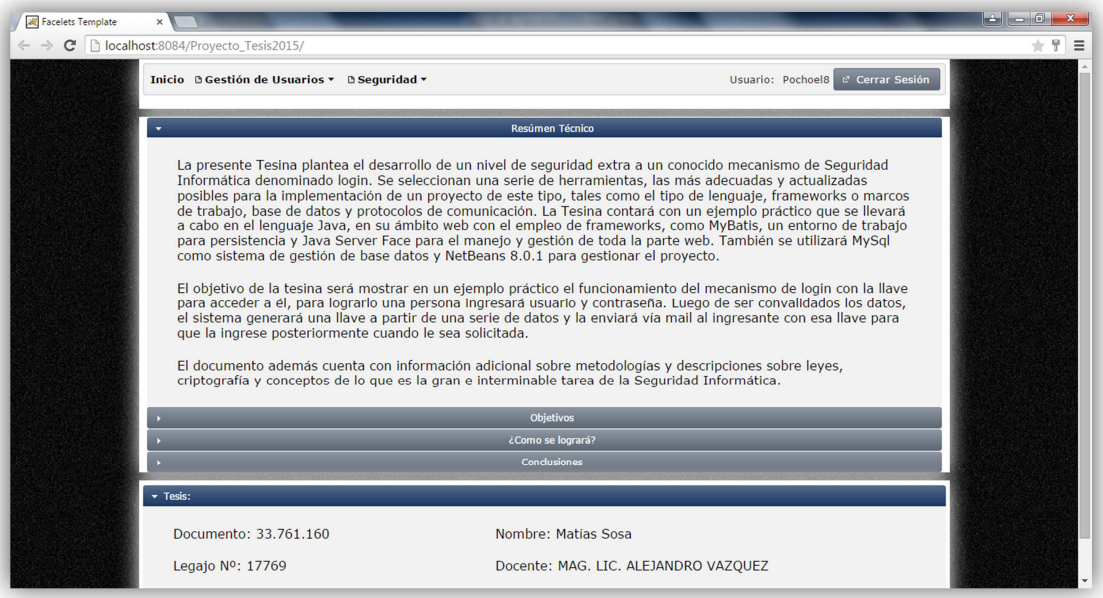


Captura de una solicitud de confirmación que aparece en pantalla.

Si el usuario decide dar NO por respuesta, no se efectuará el login y no se ejecutarán los procesos posteriores del cálculo de hash y envió de mail.

3.4.6.1.2.Página: Principal

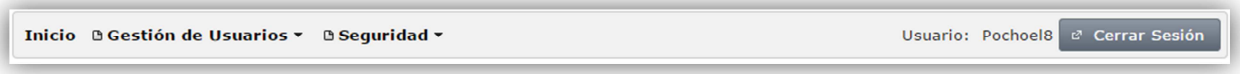
La pantalla principal está compuesta por elementos importantes en el documento de la tesis mostrando en la pantalla el resumen técnico, objetivos, como se logrará y a qué conclusiones se llagaron.



Captura de la página principal del sistema

3.4.6.1.3.Menú del Usuario

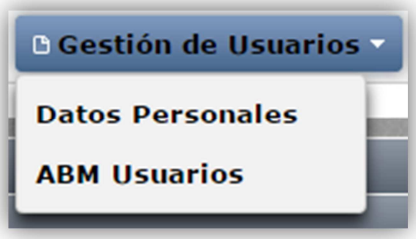
Este menú está compuesto por dos partes: la primera cuenta con 3 elementos de Menú: Inicio, Gestión de Usuarios y Seguridad. Los cuales pueden ser visualizados por todos sus usuarios. Y la otra parte, que se posiciona a la derecha cuenta con un panel donde se observa al usuario y un botón para cerrar la sesión.



Captura del Menú General del Sistema

3.4.6.1.3.1. Ítem de Menú: Gestión del Usuario

Este menú contiene 2 subelementos: Datos Personales y ABM Usuarios (Alta, Baja y Modificación de Usuarios).



Captura del Menú Gestión de Usuario

3.4.6.1.3.2. Ítem de Menú: Seguridad

Este menú contiene 2 subelementos: Perfiles y Ver Usuarios Conectados.

Captura del Menú desplegable Seguridad

3.4.6.1.3.3. Pie de Página

El pie de página que contienen páginas del sistema (salvo la de login). Contiene los datos personales y del docente a cargo de la revisión de la tesina.

Pie de Página

3.4.6.1.4.Página: Datos Personales

La página de Datos personales muestra 3 elementos desplegables: Datos de Usuario, Datos Personales y Datos de la Sesión.

En datos del usuario el usuario podrá cambiar la contraseña (referencia a interfaz punto: 3.5.6.9).

Interfaz para Datos del Usuario

En la segunda solapa se muestran los datos personales del usuario.

Interfaz para Datos Personales

En la segunda solapa se muestran los datos de la sesión personal del usuario.

Interfaz para Datos de Sesión

3.4.6.1.5.Página: ABM Usuarios

La página ABM Usuarios contiene un listado con sus respectivos filtros de los usuarios que existen en el sistema.

Interfaz para Perfil Administrador

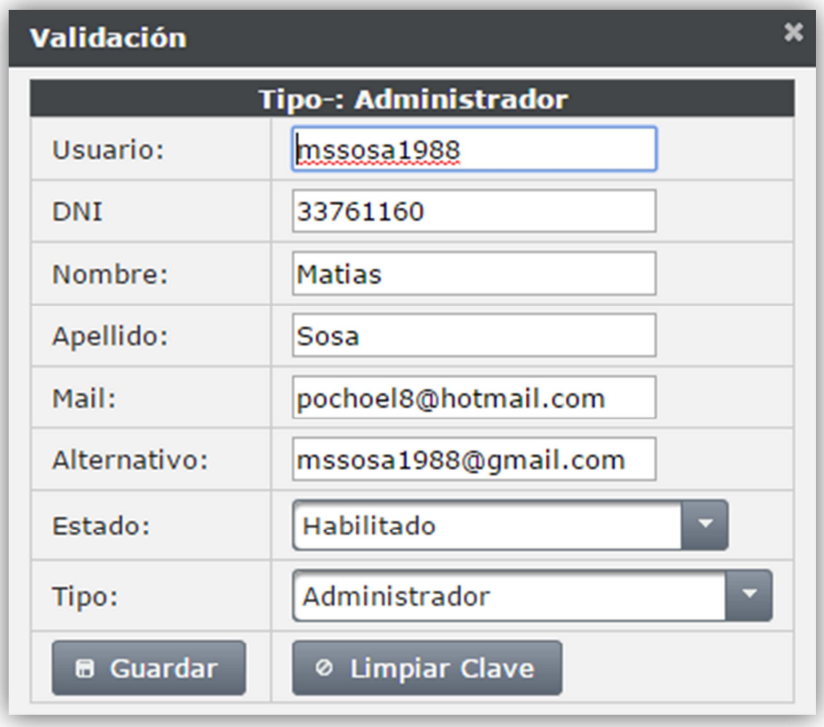
Interfaz para Perfil Supervisor

Interfaz para Perfil Operador Común

Interfaz para Perfil Consulta

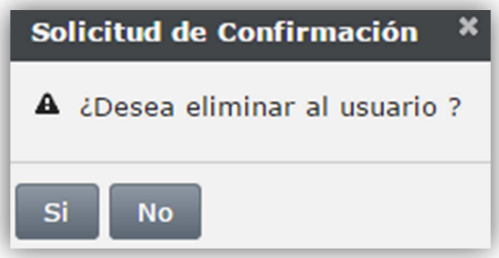
Para el usuario Operador común se puede acceder a la interfaz de Agregar Usuario.

Interfaz Agregar Usuario



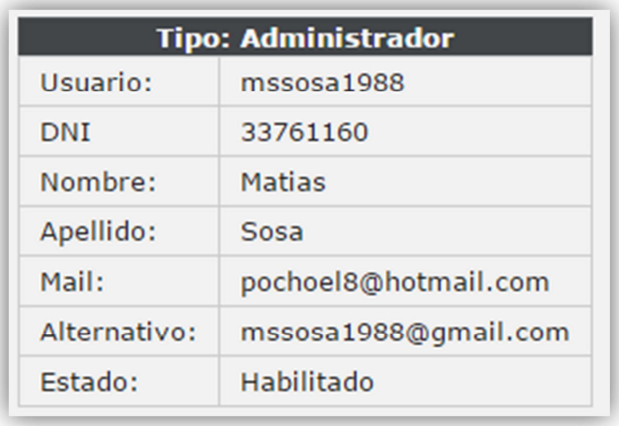
Interfaz Editar Usuario

Es un cartel MODAL que se muestra para verificar si el usuario Administrador o Supervisor.



Interfaz Confirma Eliminar Usuario

Todos los usuarios pueden ver el detalle de un usuario.

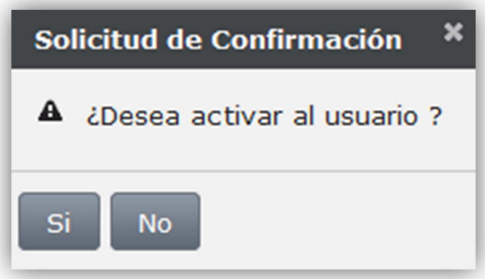


Interfaz Ver Usuario

A la interfaz de los usuarios eliminados, a la que pueden acceder solo Administradores y Supervisores.

Interfaz Ver Usuarios Eliminados

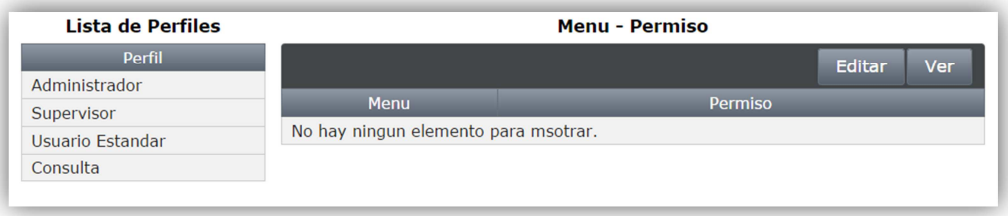
Una vez se da click en activar se muestra la ventana MODAL de confirmación.



Interfaz Confirma Activar Usuario

3.4.6.1.6.Página: Perfiles

En la página de perfiles se puede evidenciar los perfiles existentes en el sistema y el permiso que posee con respecto al menú. Un administrador o supervisor puede acceder.



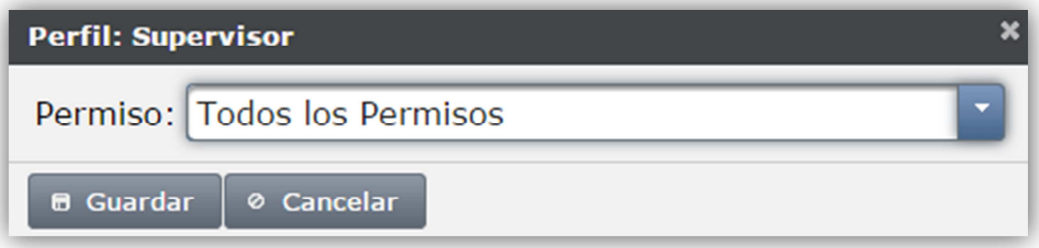
Interfaz de Lista de Perfiles y Menús

Una vez se da click en uno de los perfiles se completa la tabla de Menú-Permiso:

Interfaz de Lista de Perfiles y Menús-Permisos desplegado (Administrador)

Interfaz de Lista de Perfiles y Menús-Permisos desplegado (Supervisor)

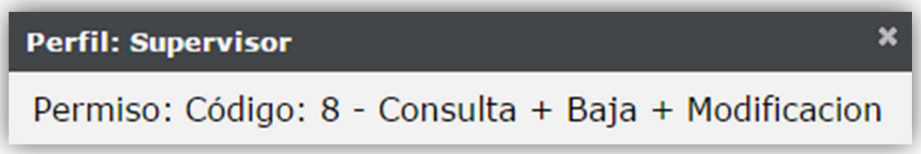
Un administrador o supervisor puede editar esos permisos mediante el siguiente menú:



Interfaz de Edición de Permiso del menú

Interfaz donde muestra las opciones de la interfaz anterior.

Un administrador o supervisor pueden dar click en ver, para que se despliegue una interfaz MODAL para ver en detalle lo expuesto en la tabla.



Interfaz Para ver en detalle un permiso de un menú

3.4.6.1.7.Página: Usuarios Conectados

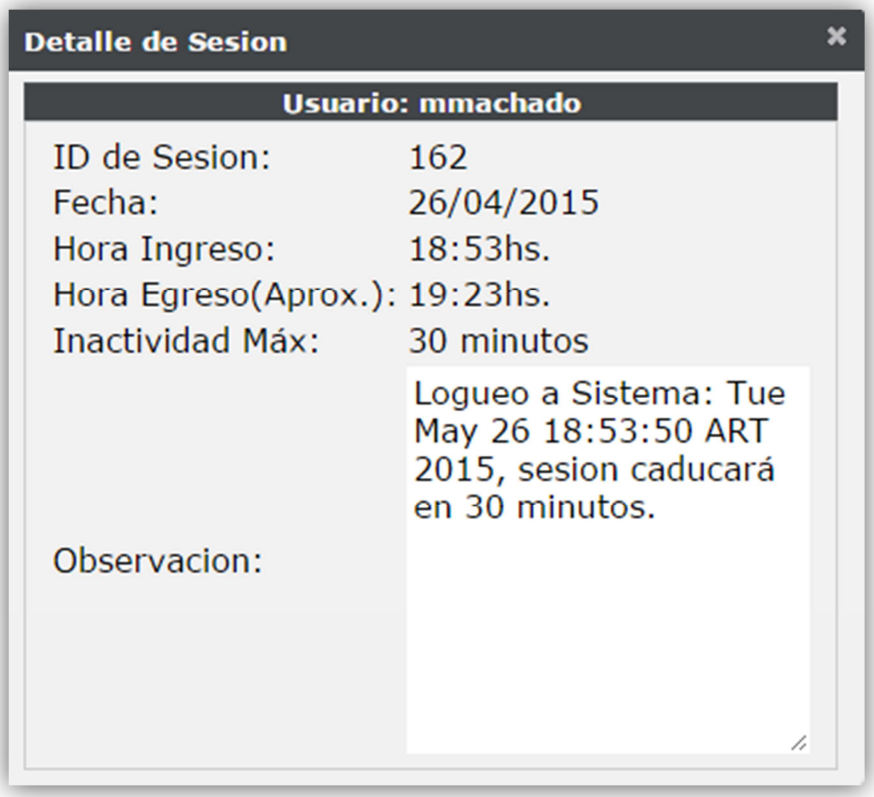
Un administrador solamente puede acceder a esta página por defecto. Luego el administrador puede elegir que un perfil determinado pueda acceder a ella.

Interfaz de Usuarios Conectados

Un administrador puede desconectar a un usuario que vea en el sistema y crea conveniente que se dé de baja su sesión. Para ello dará click en desconectar y se le aparecerá en pantalla la siguiente ventana MODAL.

Interfaz donde se solicita confirmación para desconectar a un usuario

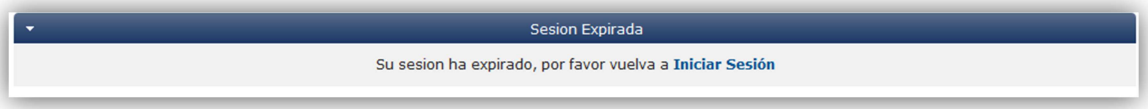
También un administrador puede ver el detalle de la sesión actual del usuario que seleccione.



Interfaz donde se visualiza el detalle de la Sesión Actual del usuario elegido

3.4.6.1.8.Página: Sesión Expirada

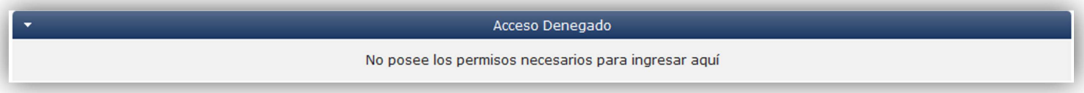
En cualquier momento que se venza la sesión o sea desconectado por un administrador el navegador lo re direccionará a una página que evidenciará el siguiente mensaje:



Interfaz Sesión Expirada

3.4.6.1.9.Página: Acceso Denegado

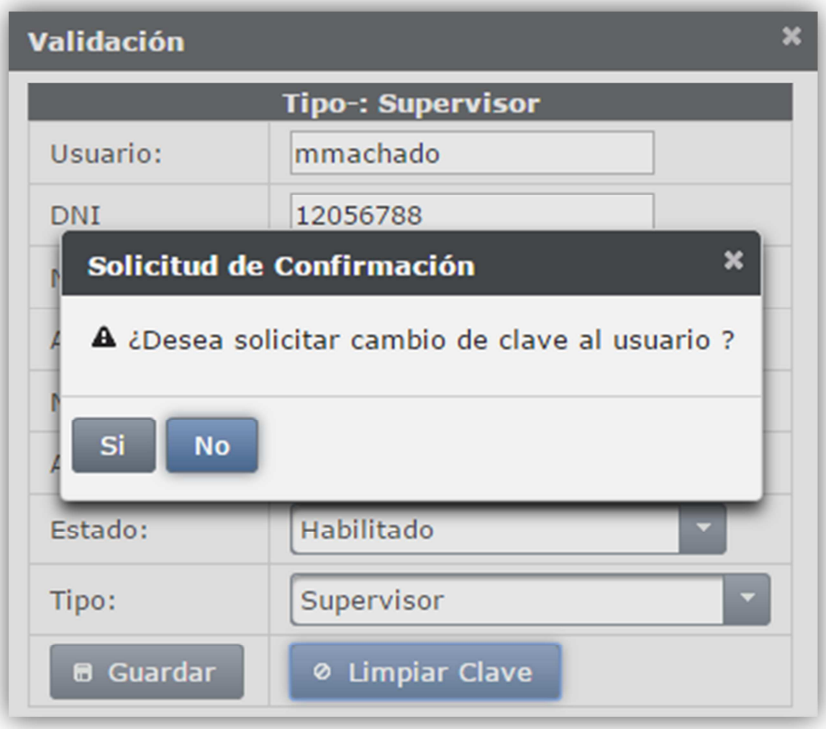
Cualquier usuario que desee acceder a un sitio que no le correspondiera se le mostrará en pantalla un cartel de acceso denegado:



Interfaz de Acceso Denegado

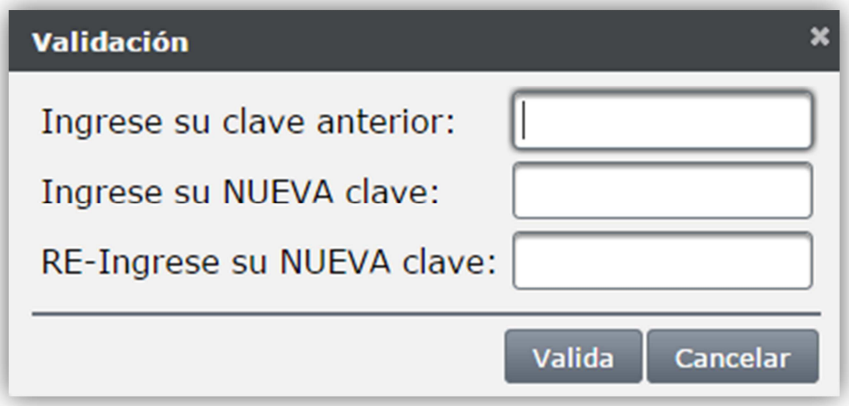
3.4.6.1.10. Solicitud de Cambio de Clave

Un administrador, operador común o un supervisor, pueden solicitar limpiar la clave del usuario. Esto hará que la próxima vez que el usuario ingrese con Usuario-Clave, le solicite el cambio de su contraseña.



Interfaz de confirmación para limpiar clave usuario.

La interfaz que se mostrará cuando un usuario ingrese sus datos y le pida el cambio de clave será:



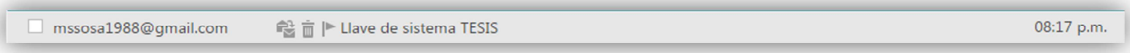
Interfaz donde usuario cambia la clave

3.4.6.1.11.Mail Enviado al Usuario

Una vez que el usuario ingresó al sistema con sus datos correctamente se le enviará un mail con el hash que contiene la llave que el sistema solicitará para completar el login. Luego se evidencia una ventana MODAL, mostrada en el punto 3.5.6.1 donde el ingresante copiará y pegará esa cadena de texto que recibió por mail.

3.4.6.1.11.1. Bandeja de Entrada

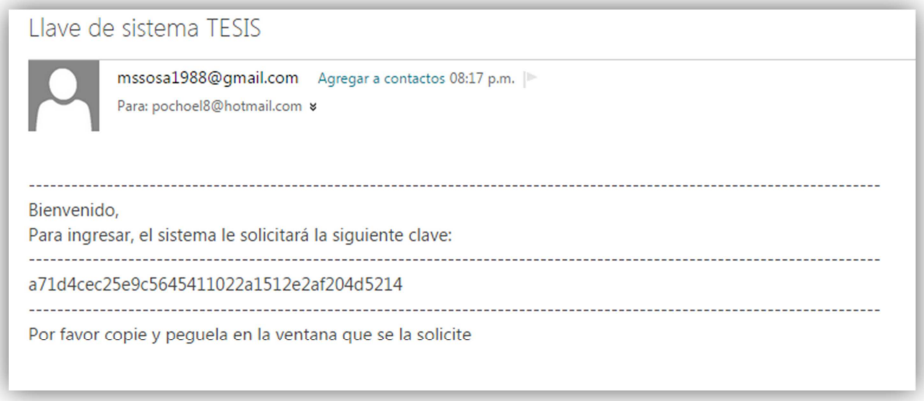
EL correo lo recibirá en su bandeja de entrada con el Asunto “Llave de sistema TESIS”



Captura de la Bandeja de entrada de Hotmail

3.4.6.1.11.2. Mensaje con Asunto: llave de sistema TESIS

El mail contendrá lo mostrado en la siguiente captura:



Captura del mail abierto en Hotmail

3.4.7.Metodología de generación del Hash tratada con ejemplo

Ejemplo: Caso con Administrador

A continuación detallaremos lo ocurrido con el sistema mediante la implementación de tablas, texto y estadísticas para mostrar como el genera el hash con un ejemplo práctico y visible, mediante el cual se podrá observar los pasos intermedios generados y documentados. Para ello tomamos el usuario administrador mssosa1988.

Primero ingresamos en el Sistema con usuario y contraseña

Usuario: mssosa1988
*Contraseña: *********

El sistema convalida la contraseña como correcta.

El sistema verifica si el usuario se encuentra online.

En caso afirmativo el sistema lo desconecta y guarda en el log, agregando que fue desconectado en hora especifica.

"Logueo a Sistema: Wed May 27 13:18:14 ART 2015, sesión caducará en 30 minutos. Esta sesión fue cerrara debido a que se produjo un ingreso desde otro navegador en fecha: Wed May 27 18:53:13 ART 2015"

El sistema armará 2 grupos para poder generar el hash:

Grupo:	Grupo 1				Grupo 2			
Identificador	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Descripción	*Nom*	*Ape*	*DNI*	*Mail*	*Hora*	*Minutos*	*Segundos*	*Fecha*

Del grupo 1 identificador 1=Nombre, 2=Apellido, 3=DNI y 4= Mail. Del grupo 2 el Identificador 1=Hora, 2=Minutos, 3=Segundos y finalmente 4=Fecha (completa incluye hora, minuto, segundo y día de la semana en inglés).

Para el ejemplo práctico:

El grupo 1 contiene datos Personales del USUARIO.

Grupo 1			
Nombre	**Apellido**	**Mail**	**DNI**
Matías	Sosa	pochoel8@hotmail.com	33761160

El grupo 2 contiene información sobre la hora/fecha del momento del login

Grupo 2			
Hora	**Minutos**	**Segundos**	**Fecha**
13	6	31	Wed May 27 13:06:31 ART 2015

Con estos datos ya almacenados en variables temporales itera 4 veces el procedimiento, eligiendo mediante la función Random () de java (acotada entre el 1 y el 4) uno de los miembros del primer grupo. Posteriormente se lanza nuevamente el método Random (), también acotada con el mismo criterio y se elige otra variable pero del segundo grupo. Y luego de esto itera nuevamente.

Veamos el ejemplo:

Operación	Iteración 1
Random(1-4)	*2*
primer_elegido	*Sosa*
Random(1-4)	*1*
segundo_elegido	*13*
concatenar	*Sosa13*
cadena_final	*Sosa13*

En la primer ejecución de Random () salió el número 2 por ende se seleccionará “Apellido” y se almacenará en la variable “primer_elegido”. Luego se lanza nuevamente la función y se obtuvo 1 por resultado. Esto quiere decir que el dato elegido es “Hora” y se almacena en la variable “segundo_elegido”.

Posteriormente de la asignación se concatenan primer_elegido y segundo_elegido para obtener por resultado “Sosa13”. Finalmente acaba la iteración cuando concatena la cadena final con las anteriores, en este caso como es la primera, queda como está.

Se comienza la iteración 2, limpiando variables y lanzando nuevamente los Random.

Operación	Iteración 2
Random(1-4) 1	*2*
primer_elegido	*Sosa*
Random(1-4) 2	*2*
segundo_elegido	*6*
concatenar	*Sosa6*
cadena_final	*Sosa13Sosa6*

En la segunda ejecución de Random () salió el numero 2 nuevamente por ende se seleccionará “Apellido” y se almacenará en la variable “primer_elegido”. Luego se lanza la función Random () para el segundo grupo y se obtuvo 2 por resultado .Esto quiere decir que el dato elegido es “Minutos” y se almacena en la variable “segundo_elegido”.

Posteriormente de la asignación se concatenan primer_elegido y segundo_elegido para obtener por resultado “Sosa6”. Finalmente acaba la iteración cuando concatena la cadena final con las anteriores, obteniendo el valor “Sosa13Sosa6” para la variable “cadena_final”.

Iteración 3.

Operación	Iteración 1
Random(1-4) 1	*1*
primer_elegido	*Matías*
Random(1-4) 2	*4*
segundo_elegido	*Wed May 27 13:06:31 ART 2015*
concatenar	*MatíasWed May 27 13:06:31 ART 2015*
cadena_final	*Sosa13Sosa6MatiasWed May 27 13:06:31 ART 2015*

Iteración 4.

Operación	Iteración 1
Random(1-4) 1	*3*
primer_elegido	*pochoel8@hotmail.com*
Random(1-4) 2	*4*
segundo_elegido	*Wed May 27 13:06:31 ART 2015*
concatenar	*pochoel8@hotmail.comWed May 27 13:06:31 ART 2015*
cadena_final	*Sosa13Sosa6MatiasWed May 27 13:06:31 ART 2015pochoel8@hotmail.comWed May 27 13:06:31 ART 2015*

Luego de terminar con la cuarta iteración se obtuvo por valor de cadena final:

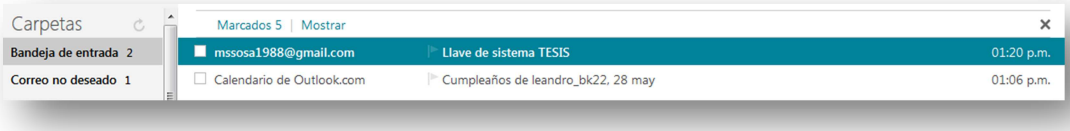
“Sosa13Sosa6MatiasWed May 27 13:06:31 ART 2015pochoel8@hotmail.comWed May 27 13:06:31 ART 2015”

Ese valor obtenido en formato de cadena de texto es procesado mediante el método de encriptación SHA-1 y se obtiene el valor:

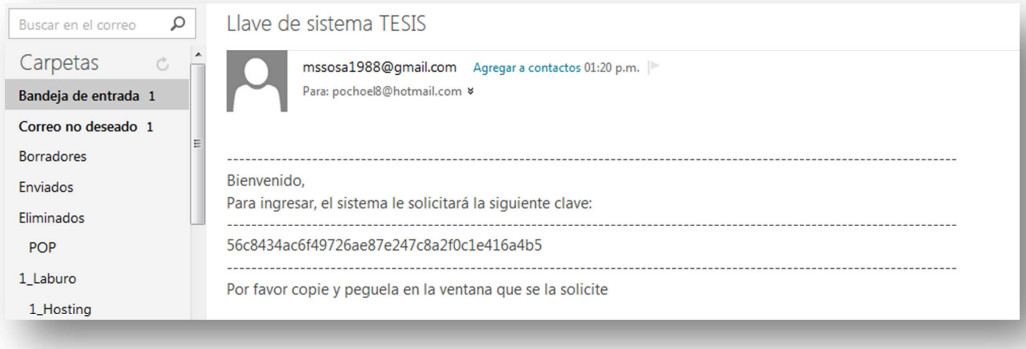
“56c8434ac6f49726ae87e247c8a2f0c1e416a4b5”

Esta es la clave encriptada que se enviará al mail para que ingrese posteriormente el usuario que desee ingresar al sistema. Como se evidencia a continuación.

Bandeja de entrada de Hotmail:



Una vez abierto el mail se presenta de la siguiente manera:



Esta es la llave UNICA e IRREPETIBLE que genera el sistema.

Si este método cayera en manos de atacantes informáticos, poco habría que temer ya que el método es lo suficiente robusto que inclusive publicándolo y otorgándole usuario, contraseña y los datos de que necesitase para calcular el hash, es probabilísticamente casi imposible que lo pueda resolver exactamente en los 3 intentos que posee antes de se bloquee el usuario. La única manera es que este individuo poseyera acceso a la cuenta de correo del usuario, en tal caso poseería el hash generado.

¿Por qué es casi imposible que un cracker logre encontrar el hash?

Según la rama de combinatoria de la matemática[29] tenemos:

$$VR_{m,n} = m^n$$

VR= Variaciones con repetición.
m=4 (iteraciones)

[29] http://es.wikipedia.org/wiki/Combinaciones_con_repetici%C3%B3n#Definici.C3.B3n

n=8 (elementos)

Teniendo en cuenta que son 4 veces que se itera y las agrupaciones que haríamos son de 8 elementos obtenemos reemplazando en la fórmula:

VR = 4^8= 65.536

Entonces son 65.536 combinaciones diferentes podemos llegar a tener distribuidas en las 4 iteraciones. Lo que hace un número grande de posibilidades.

Si a esto le sumamos los 3 intentos que hay para resolverlo, esto quiere decir que la probabilidad de que algún cracker pueda romper la seguridad de la llave es inferior al 1%.

Por último para calcular las probabilidades que se mencionaron anteriormente utilizamos:

Probabilidad (A) = (Número de casos favorables)/ (Número de casos posibles)[30]

	Intento 1	**Intento 2**	**Intento 3**
Casos Favorables	*1*	*1*	*1*
Casos Posibles	*65536*	*65535*	*65534*
Total	*0,000015259*	*0,000015259*	*0,000015259*
%	*0,001525879*	*0,001525902*	*0,001525925*

El recuadro anterior nos detalla los intentos que poseerá el cracker para acceder al HASH generado en SHA-1. En caso de fallarla por tercera vez consecutiva el sistema bloqueará al usuario y no lo dejará validar usuario y contraseña hasta que algún administrador o supervisor cambie su estado y lo coloque en habilitado nuevamente.

Como se puede observar la probabilidad de acierto es muy baja debido a la cantidad inmensa de combinaciones que pueden tomar el algoritmo y a la escasísima cantidad de intentos asignada por sistema.

[30] http://www.ub.edu/stat/GrupsInnovacio/Statmedia/demo/Temas/Capitulo1/B0C1m1t5.htm

4. CONCLUSIONES

En la etapa final del trabajo, corresponde realizar un análisis para poder observar si se alcanzó el objetivo planteado y poder obtener una conclusión final.

Se utilizaron tecnologías Java, Mysql y dos Frameworks PrimeFaces y MyBatis para el desarrollo del ejemplo demostrativo de la tesina, para así, cumplir con los objetivos de:

- **ACCESO ÚNICO AL SISTEMA**

- **AGREGAR UNA CAPA EXTRA DE AUTENTICACIÓN A LA SEGURIDAD DEL SISTEMA.**

- **ENCRIPTAR LA CONTRASEÑA ENVIADA AL CORREO ELECTRÓNICO.**

El sistema desarrollado durante la tesina dista en grandes pasos de ser un sistema profesional, por motivos de que si bien puede utilizarse de base para arrancar con uno de mayor envergadura, para alcanzar el estatus de profesional requeriría informes, consultas avanzadas parametrizables, parámetros de configuración entre otros innumerables requisitos que puede llegar a necesitar un cliente. Pero, para demostrar la casuística alcanza con el ejemplo planteado donde solo hay una registración de usuario que la pueden hacer solo los usuarios estándares y administradores y unas consultas simples.

El uso de tecnologías de vanguardia, hace que sea un sistema que posee características que pueden modificarse y actualizarse gradualmente y de manera mucho más sencilla. Ya que continuamente se avanza en versiones y documentación generada por la comunidad. También esto se vuelve una desventaja ya que no todos los que aportan a la comunidad son expertos, al contrario, muchos son amateur o nivel medio que encuentran soluciones fáciles copiadas de otro y las suben como “la novedad”. En esta casuística es donde pueden encontrarse errores groseros o brechas de seguridad muy grandes, sumado al avance de las tecnologías que constantemente cambian y mutan en torno a la evolución natural. Producto de que salen al mercado nuevos lenguajes de

programación, nuevas metodologías, frameworks y sobretodo tecnologías diferentes con las cuales interactuar.

En instituciones como la ATM (Administración Tributaria Mendoza) puede llegar a ser importante el uso de la metodología planteada en esta tesina ya que, ante la ausencia - por motivos que fuere - de la utilización de firma digital, es un buen substituto para que se puedan iniciar trámites de uso interno con sistemas informatizados, que, en la práctica se generan con papel y documentación creando la dependencia, sumado a todos los inconvenientes que conlleva su uso.

Por lo tanto si se implementa una metodología similar, con las bases planteadas o inclusive la misma que se plantea en la tesina, se puede lograr el paso inicial para la documentación digital gubernamental en Mendoza o algunas áreas que requieren la optimización del papel a un medio más acorde a la época. Después se plantearán mejoras y tal vez termine siendo –lógicamente- algo obsoleto, pero siempre es recomendable empezar y con tanta impedimenta burocrática en el medio es una buena solución de compromiso ya que debemos recordar que lo perfecto es inalcanzable en tiempos finitos.

Para finalizar se hace especial hincapié en el capítulo 3.6 de este documento, donde se muestran valores estadísticos y se refleja que aun entregando el algoritmo y los datos a un atacante experto (cracker) para poder realizar la llave del sistema se tiene una probabilidad menor al 1% de averiguar la clave en los 3 intentos que posee antes de que se bloquee el usuario. Por lo tanto el algoritmo está dotado de cierta robustez para su implementación y perfectamente puede ser publicado, ya que la llave se genera aleatoriamente y su publicación no sería una brecha de seguridad. Notoriamente si el sistema sufre algún ataque de otro tipo, la llave no tendrá nada que hacer, por ende se cumple la premisa de que ningún sistema es 100% invulnerable.

Para resumir y redondear la idea de las conclusiones, a partir de lo expuesto anteriormente, podemos decir que se cumple con los 3 objetivos planteados. Ya que posee acceso único al sistema, porque el mismo se desarrolló partiendo de la seguridad en donde se verifica en todo momento la sesión del usuario. También se cumplió el objetivo de agregar una capa extra de autenticación a la seguridad del sistema, empleando un algoritmo de desarrollo propio con una clave difícilmente alcanzable. Inclusive, si un atacante profesional contara con la información de la sesión y los datos personales con la que se genera el hash, sería altamente dificultoso que alcance la contraseña generada por el método. Ya que una vez generada y encriptada se envía al correo electrónico cumpliendo con el tercer objetivo citado al comienzo de la conclusión. En caso de existir algún problema de conectividad, el sistema NO podrá enviar la clave y el usuario se quedará sin acceder al mismo.

5. BIBLIOGRAFÍA

Para el desarrollo de la tesina se consulta la siguiente bibliografía:

AGUILERA LOPEZ (2010) Seguridad Informática, Editorial Editex S.A. 240 págs.

ASP.NET. (2014) ¿Qué es ASP.NET? Recuperado el 05 de mayo de 2014, de http://mysf.galeon.com/segunda.htm#¿Qué es ASP.NET

BASE DE DATOS. (2014, 26 de mayo) Wikipedia, La enciclopedia libre. Recuperado el 1 jun 2014, 00:59 http://es.wikipedia.org/w/index.php?title=Base_de_datos&oldid=82738614>.

BELÓN PEREZ (2014) ¿Cómo elegir un servidor Web?. Recuperado el 15 de mayo de 2014, de http://www.programadorphp.org/blog/como-elegir-servidor-web/

BLOGS UTPL (2009) Ventajas y desventajas del PHP2. Recuperado el 01 de mayo de 2014, de http://blogs.utpl.edu.ec/disenowebymultimedia/2009/07/23/ventajas-y-desventajas-de-php-2/

CARRION, HUGO DANIEL (2001) "Presupuestos para la Punibilidad del Hacking". Tesis. Recuperado el 13 de julio de 2014, de http://www.delitosinformaticos.com/tesis.htm

CLR (2015) Common Language Runtime. Recuperado el 01 de mayo de 2014, de http://msdn.microsoft.com/es-es/library/8bs2ecf4(v=vs.110).aspx

COMBINACIONES CON REPETICIÓN. (2014, 26 de diciembre). Wikipedia, La enciclopedia libre. Fecha de consulta: 01:17, junio 1, 2014 http://es.wikipedia.org/w/index.php?title=Combinaciones_con_repetici%C3%B3n&oldid=79005898

DEVELOPER NETWORK (2014) ¿ASP.NET es gratis o tiene algún costo? Recuperado el 06 de mayo de 2014, de http://social.msdn.microsoft.com/Forums/es-ES/e50e3d1d-6dc5-4521-a973-8f4d41544000/aspnet-es-gratis-o-tiene-algn-costo?forum=netfxwebes

FRAMEWORK. (2015, 19 de marzo). Wikipedia, La enciclopedia libre. Fecha de consulta: 13:51, mayo 31, 2014 desde http://es.wikipedia.org/w/index.php?title=Framework&oldid=80876102.

IBM INFORMIX (2015) An Intelligent database for Internet of Things. Recuperado el 15 de mayo de 2014, de http://www-01.ibm.com/software/data/informix/

INNOVACION UB (2015) ¿Cómo se calculan las probabilidades? Recuperado el 26 de mayo de 2014, de http://www.ub.edu/stat/GrupsInnovacio/Statmedia/demo/Temas/Capitulo1/B0C1m1t5.htm

JAVA DB (2015) Java SE Technologies – Database. Recuperado el 05 de mayo de 2014, de http://www.oracle.com/technetwork/java/javase/jdbc/index.html

JAVA. (2015) Qué es Java. Características de Java como lenguaje de programación. Recuperado el 02 de mayo de 2014, de http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html

JCSCR (2012) Paper Journal of Computer Science & Research - ISSN 2227-328X Vol. 1, No. 1, Pages. 20-31, Recuperado el 16 de abril de 2014, de http://www.jcscr.com

LIMA DE LA LUZ, MARÍA (1984) Criminalia N° 1-6 Año L. Delitos Electrónicos. Ediciones Porrua. México.

LINKEDIN (2014) Ventajas y desventajas de hacer web en ASP.NET. Recuperado el 06 de mayo de 2014, de http://www.linkedin.com/company/1078501/advantages-disadvantages-of-making-website-in-asp-net-537455/product

MAIORANO, A. (2009) Criptografía, Técnicas de desarrollo para profesionales. México D.F.: ALFAOMEGA GRUPO EDITOR, S.A.

MOLINER, MARÍA (1996) Diccionario de María Moliner Edición Digital. Copyright© Novel Inc.; Copyright © Maria Moliner.

MONITOR (SYNCHRONIZATION). (2015, May 31). In Wikipedia, The Free Encyclopedia. Retrieved 00:14, January 10, 2015, from http://en.wikipedia.org/w/index.php?title=Monitor_(synchronization)&oldid=664850637

NET FRAMEWORK 3.0 (2015) Elemento process Model (Esquema de configuración de ASP.NET).Recuperado el 05 de mayo de 2015, de http://msdn.microsoft.com/es-es/library/7w2sway1(v=vs.85).aspx

ORACLE (2015) Oracle Base de Datos 11g. Recuperado el 06 de mayo de 2015, de http://www.oracle.com/es/solutions/midsize/oracle-products/database/index.html

PHP.NET (2015) Manual del PHP. Referencias del lenguaje. Clases y objetos. Recuperado el 02 de mayo de 2015, de http://www.php.net/manual/es/oop5.intro.php

PRINCIPIO DE SUBSIDIARIEDAD. (2014, 26 de septiembre). Wikipedia, La enciclopedia libre. Fecha de consulta: 14:25, mayo 31, 2015 desde http://es.wikipedia.org/w/index.php?title=Principio_de_subsidiariedad&oldid=77192836.

SCRIBD.ES (2015) Tabla de comparación de los diferentes DBMS. Recuperado el 15 de mayo de 2015, de https://es.scribd.com/doc/109205352/Tabla-de-comparacion-de-los-diferentes-DBMS

TÉLLES VALDEZ, Julio (1996). Derecho Informático. 2° Edición. Mc Graw Hill. México. Pág. 103-104

TORRES, SILVIA (2013)La cita y la referencia bibliográfica: guía basada en las Normas APA. UCES. Recuperado el 30 de mayo de 2015, de http://www.uces.edu.ar/biblioteca/citas-bibliograficas-APA-2012.pdf